

NASA CR-139147

DOCUMENTATION OF THE DATA ANALYSIS SYSTEM  
FOR THE GAMMA RAY MONITOR ABOARD OSO-H

(NASA-CR-139147) DOCUMENTATION OF THE  
DATA ANALYSIS SYSTEM FOR THE GAMMA RAY  
MONITOR ABOARD OSO-H Final Project  
Report (New Hampshire Univ.) 369 p HC  
\$10.00

N75-10976

Unclass

CSSL 05B G3/15 02588

S. CROTEAU, A. BUCK, P. HIGBIE, J. KANTAUSKIS,  
S. FOSS, E. CHUPP, D. FORREST, A. SURI, I. GLESKE

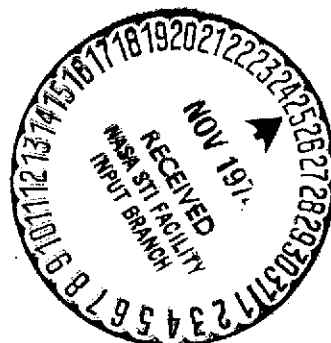
DEPARTMENT OF PHYSICS  
UNIVERSITY OF NEW HAMPSHIRE  
DURHAM, NEW HAMPSHIRE 03824

DECEMBER 1973

FINAL PROJECT REPORT FOR NAS 5-11054  
DOCUMENTING DATA ANALYSIS SYSTEM

PREPARED FOR

NATIONAL SPACE SCIENCE DATA CENTER  
GODDARD SPACE FLIGHT CENTER  
GREENBELT, MARYLAND 20771



DOCUMENTATION OF THE DATA ANALYSIS SYSTEM  
FOR THE GAMMA RAY MONITOR ABOARD OSO-H

S. CROTEAU, A. BUCK, P. HIGBIE, J. KANTAUSKIS,  
S. FOSS, E. CHUPP, D. FORREST, A. SURI, I. GLESKE

DEPARTMENT OF PHYSICS  
UNIVERSITY OF NEW HAMPSHIRE  
DURHAM, NEW HAMPSHIRE 03824

DECEMBER 1973

FINAL PROJECT REPORT FOR NAS 5-11054  
DOCUMENTING DATA ANALYSIS SYSTEM

PREPARED FOR

NATIONAL SPACE SCIENCE DATA CENTER  
GODDARD SPACE FLIGHT CENTER  
GREENBELT, MARYLAND 20771

## Table of Contents

	<u>Page</u>
Preface	i
Section 1 - Subprogram FEEDER . . . . .	1-1
Section 2 - PASS-I. . . . .	2-1
Section 3 - CAL1. . . . .	3-1
Section 4 - CAL2. . . . .	4-1
Section 5 - PASS-III. . . . .	5-1
Section 6 - Van Allen Belt Predict Program. . . . .	6-1
Section 7 - University of New Hampshire Computation Center PLOT Routine . . . . .	7-1
Section 8 - Response Function Programs. . . . .	8-1

## Preface

This report contains a description of the programming system developed to prepare the data from the UNH (University of New Hampshire) gamma ray monitor on OSO-7 for scientific analysis. In order to help understand the reasons for our particular type of program development, this short preface will describe, from a scientific point of view, the detector, the data, and the objectives of the scientific analysis.

The gamma ray monitor consisted of a 7.6 cm x 7.6 cm NaI scintillator-photomultiplier central detector. This detector was covered on all sides, except for a forward aperture, by a 5 cm thick CsI scintillator shield. This shield acted as a collimator for gamma rays and also rejected direct charged-particle counts. The instrument was on the wheel section of OSO, which was spinning in a plane that always included the sun with a 2-second period. Data was taken during two 0.5-second intervals when the detector was "looking" at the sun and directly away from the sun. These two intervals were called the "solar" and "background" quadrants.

A small X-ray detector was also included in the instrument and data was taken from this in the same two quadrants.

The most important information from the monitor consisted of the two 370-channel pulse height spectra. Data in these spectra was accumulated over a 180-second time interval when the detector was pointing in the solar and background quadrant. Analysis of this data required that the channel numbers in the pulse height spectra be calibrated with an incident gamma ray of known energy. This calibration was accomplished twice per orbit when a radioactive source (RS cal) was gated into the detector instead of the normal spectrum. Therefore, the sets of pulse height spectra, the RS calibration spectra, counting rates from the various shield elements and the X-ray detector, and house-keeping measurements constitute the data that must be analyzed.

Data from the experiment was received in two ways: "quick look" data and experimenter tapes. "Quick look" data consisted of raw data received from a few selected orbits each day and was promptly sent to UNH via a telephone line. The experimenter tapes consisted of semiprocessed and cleaned-up data containing all of the information received from the experiment. The experimenter tapes were the prime data source for all scientific analysis.

Two pieces of hardware were used to process this data: a Varian 620/i mini-computer and the University's IBM-360-H50 computer. The Varian 620/i was used to process the quick look data for operational and engineering uses. As we will see below, it is used in the ongoing analysis of the fully processed prime data from the experimenter tapes. The IBM-360-H50 was used to process the experimenter tapes and to prepare new tapes containing data which could be more closely analyzed on the Varian. This report contains a description of the final procedure and programs used to reduce the original experimenter data.

The experimenter tapes were processed in several passes. These include:

- Pass-I - strips off the X-ray and raw main frame (MF or central detector) data for use on the Varian 620. It also removes the calibration (RS) spectrum and provides various engineering and operational outputs.
- Cal 1 - processes the RS calibration spectra for visual inspection and for use in Cal 2.
- Cal 2 - provides a continuous gain calibration function versus time from Cal 1 and a manual input. This output allows us to calibrate all the central detector spectra that were taken between the RS calibrations.
- Pass-III - provides standardized and gain-corrected spectra for use on the Varian 620. Numerous geophysical and S/C parameters are also added to the data in this pass.

Subprogram Feeder was used to read the experimenter tapes in Passes I and III.

The magnetic tape output of Pass-III constitutes the prime data bank that will be used in most of the scientific analysis efforts with the data from the gamma ray monitor. This scientific analysis, in brief, consists of selectively summing together numerous spectra in a search for gamma ray sources and events. The selection process consists of using only data in the proper time interval and with the proper geophysical and S/C parameters. The objective of the selection process is to maximize the gamma ray signal while minimizing the gamma ray background. As mentioned before, the selection process is carried out on the Varian 620/i computer.

The programs described in this report are the result of several false starts and many modifications. These final programs, however, have accomplished their aim very well indeed as the current analysis efforts using the final outputs continue to show.

David J. Forrest  
Principal Scientist

## Section 1

### Subprogram FEEDER

	<u>Page</u>
Subprogram FEEDER	1-1
Subroutine ADRESS	1-58
Subroutine DECISION	1-59
Subroutine GETCOR	1-60
Subroutine HEADEREC	1-61
Subroutine ORBSTOR	1-62
Subroutine ORBWNDW	1-63
Subroutine TAPREAD	1-66
FEEDER Messages	1-68

## Subprogram FEEDER

### Table of Contents

- I. Introduction
- II. Introductory View of Subprogram FEEDER
  - A. Summary of Decomm Tape organization
  - B. Summary of Data Supplied by FEEDER
  - C. Summary of Commands FEEDER can respond to
- III. Suggested Variable Names Invented for this Report
- IV. Detailed Descriptions of the Data Supplied by FEEDER
  - A. HEADER Window
  - B. ORBIT Window
  - C. DECISION Window
  - D. SCAN Window
  - E. INTERRUPTION Vector
- V. Detailed Description of Commands FEEDER can Respond to
  - A. Initialization
  - B. Advancing the Decomm Tape by Half-blocks
  - C. Retrieving Orbit data
  - D. Retrieving Decomm File Label Information
  - E. Revising the DECISION window
  - F. Revising Main Frame Data
  - G. Deleting Half-blocks from the Data Stream
  - H. File Skipping and Tape Skipping
  - I. Retrieving Main Frame Data



## Subprogram FEEDER

## I. Introduction

## A. Assumptions that influenced the organization of this report

It is assumed this report will primarily be used as a reference by individuals concerned with developing programs to analyze data from the UNH gamma ray monitor on the OSO-H spacecraft. Such individuals are assumed to be familiar with the Fortran programming language, and thus will recognize the meaning of the Fortran expressions used in this report to define the output of subprogram FEEDER. Also assumed is cognizance of the types of data produced by the UNH gamma ray monitor, of the manner these data are read onto the spacecraft's tape recorders, and of the types of auxiliary information merged by GSFC processing of OSO-H data.

This report is presented in order of increasing detail. The balance of this introduction reviews the objectives of subprogram FEEDER. Section II summarizes the categories of FEEDER's output as well as the types of commands a calling program must issue to obtain this output. The same information is provided in more detailed form in Sections IV and V.

## B. Motivation for creating subprogram FEEDER

Experimenter Decomm tapes provided by GSFC are not conveniently organized for output to Fortran programs executed by an IBM-360 computer. About 15 percent of the words on the Decomm tapes are subdivided into bit-length fields. To separate these fields, the O/S Assembler language is easier to code and develops a more efficient program. But perhaps the major inconvenience stems from the conflict between the octal organization of data on the Decomm tapes and the hexadecimal organization of IBM-360 computers. Floating point words on the Decomm tapes have the 36-bit IBM-7094 format. Short integer words on the tapes have only 12 bits, as opposed to 16 bits for an IBM-360. When a Decomm tape is read by an IBM-360, the choice is between the evils of word misalignment and of word fragmentation. These inconveniences are compounded by the word fragmentation built into the read-out mechanism of the UNH gamma ray experiment package.

Subprogram FEEDER is designed to cope with many of the difficulties that result from the inconvenient format of Decomm tapes. Bit-length fields are separated. Floating point words are converted to one of the IBM-360 formats. Experiment word fragments are spliced together and realigned to meet IBM-360 requirements.

### C. Guidelines used in developing subprogram FEEDER

A major consideration in designing subprogram FEEDER was maximizing the ease of changing programmed strategies to identify, to interpret, to accept for analysis or to edit data from the Decomm tapes. All of these judgment-making activities are necessarily performed to some degree by any program analyzing Decomm data. To obtain data from the tapes, these analysis programs will presumably utilize subprogram FEEDER, a moderately lengthy set of routines written in Assembler language. To facilitate changing tactics, FEEDER simply stays out of the judgment-making process to the extent possible. FEEDER deletes nothing from the UNH data or from any identifiers, flags, or associated parameters that pertain to UNH data words. (Sail subcom data and non-UNH data from the digital and wheel analog subcoms are the only information always deleted by FEEDER.) Although this design feature places the burden of judgment-making solely on programs calling FEEDER, these programs will presumably be written in Fortran and thus be easier to modify.

Minimizing CPU time was another consideration in designing subprogram FEEDER. The use of Assembler language in itself allows a major reduction of CPU time. In addition FEEDER has built-in options that allow it to run at reduced work loads by ignoring types of data not needed by a calling program.

FEEDER performs a variety of services intended to help designers of calling programs. Data observation times are presented as a continuous parameter (seconds of century) to remove the necessity of constantly checking for end-of-day condition or for end-of-year condition. FEEDER constantly checks the data stream for discontinuities and indicates their presence or absence to calling programs. Other interruptive conditions are monitored and their existence reported to calling programs. Although FEEDER avoids determining whether and how to edit the data, it is designed to cooperate in certain editing activities. In addition, FEEDER performs a major re-shuffling of data from Decomm tapes in an effort to more

closely simulate the manner information is read out onto the spacecraft's tape recorders. Information from file label blocks and from orbit data blocks are shunted into storage areas while the Decomm data blocks are divided into "half-blocks" representing major frames. Consequently, the data stream appears to calling programs to be a sequence of major frames, with pertinent file label information and pertinent orbit data available on request.

## II. An Introductory View of Subprogram FEEDER

### A. Summary of Decomm tape organization

Each recording (via telemetry) of data stored on one of the spacecraft's tape recorders will normally produce one file of data on an experimenter's Decomm tapes. The quantity of data in any one Decomm file is thus restricted to the capacity (about 108 minutes) of the spacecraft's tape recorders. Decomm tapes sent to experimenters will typically contain several files of data. Presumably these files will be in chronological order.

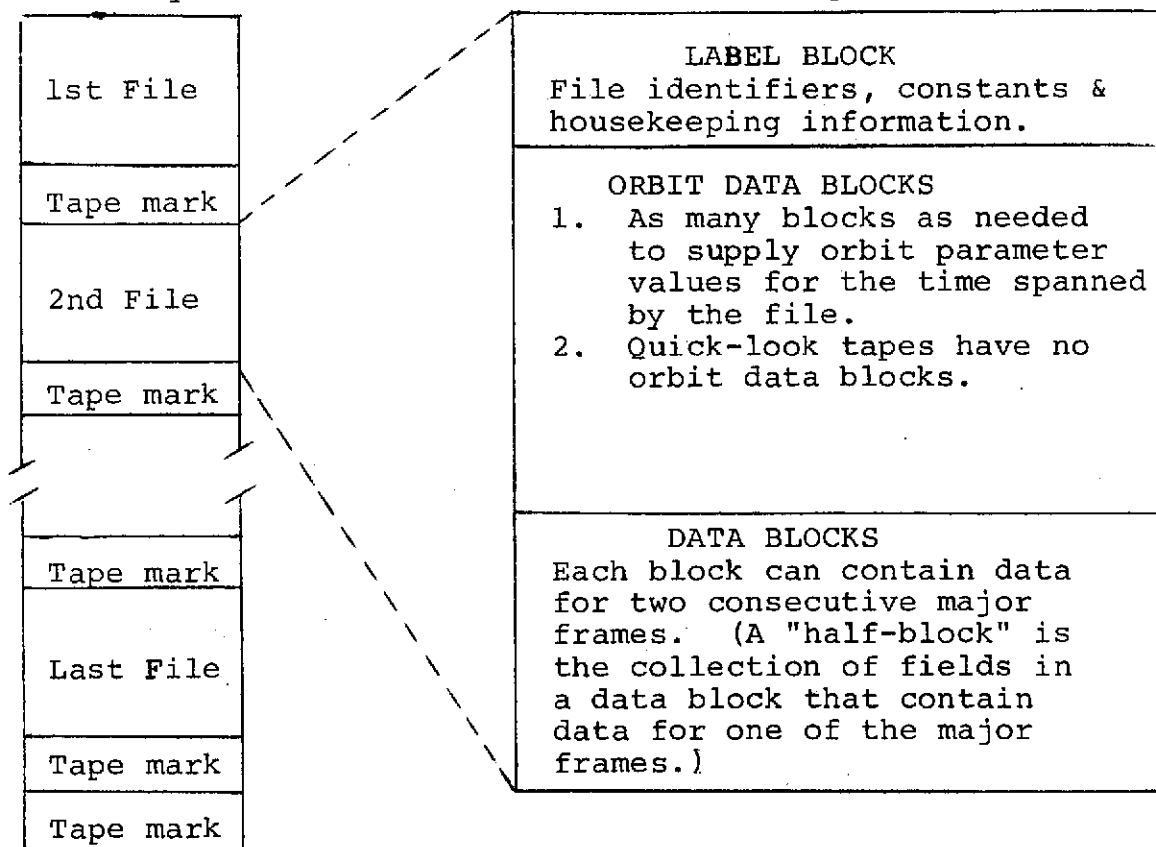


Fig. II.A.1  
Organization of a  
Decomm Tape Volume

Fig. II.A.2  
Arrangement of Data Blocks  
Within a File of a Decomm Tape

Figures II.A.1 and II.A.2 (above) show the data stream encountered by reading forward on a Decomm tape. Subprogram FEEDER rearranges the data stream to appear as a series of half-blocks. The first half-block from a file is indicated by a flag and the first half-block from a tape is indicated by an additional flag. Upon request FEEDER can supply information from either of the last two file label blocks it has read. FEEDER can also retrieve orbit data values upon request.

## B. Summary of data subprogram FEEDER supplies to a calling program

Subprogram FEEDER supplies information to a calling program through 5 named COMMON blocks referred to in this report as "windows".

### 1. The HEADER window

COMMON/HEADER/list

When requested, FEEDER will supply information about a particular file in the common block denoted by the symbol HEADER. Information can be supplied for either the file currently being read by FEEDER or the immediately preceding file, depending upon which is requested. The information supplied for a particular file includes

- a. All information contained in the label record.
- b. A list of sun-entrance and sun-exit times taken from orbit data items supplied for these events.
- c. Identifiers, such as tape volume name, file count.
- d. Time conversion constants, which can assist a calling program in computing the time of day for a particular data entry. These are also the constants used by FEEDER to compute data times.
- e. Calendar form of the begin time of file and the end time of file. These could be useful identifiers for a calling program's printout.
- f. Time(s) of ascending node crossing(s) obtained by scanning orbit data records.

### 2. The ORBIT window

COMMON/ORBIT/list

When requested, FEEDER will supply, for a given time  $t$ , orbit parameter values for times  $t_1$ ,  $t_2$ ,  $t_3$ , and  $t_4$  such that  $t_1 < t_2 < t < t_3 < t_4$ . A vector of 4 values is returned for each of the orbit parameters. Ordinarily the times  $t_1$ ,  $t_2$ ,  $t_3$ , and  $t_4$  will be one minute

apart. It is possible, of course, for the calling program to request orbit data for a time  $t$  that is either near an extreme limit of available data or outside the range of available data. In the former case, FEEDER will post a flag indicating the times of the data returned are "off center" ( $t < t_2$  or  $t > t_3$ ). In the latter case, an "out-of-range" flag is posted and the contents of the ORBIT window are left unmodified.

As FEEDER encounters the start of a new file of data, it updates the contents of the orbit data storage area it manages. However, under normal conditions, enough orbit data from the previous file is retained to accommodate any requests the calling program is likely to make.

### 3. The DECISION window

COMMON/DECIDE/list

Most of the varieties of data from Decomm data blocks are supplied to the calling program via the DECISION window. Main frame data (other than the 1st, 2nd, and 60th of the 64 experiment words) are the only type of data not included in the DECISION window.

The DECISION window contains 30 "panes". Unless modified by the calling program, each pane contains information from one half-block on the Decomm tape. Consecutive half-blocks, including all fill data half-blocks, are represented by consecutive panes of the DECISION window. The DECISION window is slightly longer than the minimum 24 panes required to allow correlation of data accumulated during corresponding time intervals. For example, if  $T_1$  and  $T_2$  are the begin time and end time for accumulating a full scan, then panes  $N - (N+11)$  might contain subcom data and attitude data read out between  $T_1$  and  $T_2$ , while panes  $(N+12) - (N+23)$  should correspond to half-blocks containing the main frame data accumulated between  $T_1$  and  $T_2$ .

Panes 15-30 correspond to half-blocks for which FEEDER can supply main frame data on request. FEEDER does not keep main frame data corresponding to panes 1-14 of the DECISION window.

Types of data in each pane of the window include:

a. Pane identifiers

- (1) Starting time (seconds of century) of the major frame, if any, represented by the pane.
- (2) Pointers to the Decomm tape block from which the pane originated.
- (3) A set of flags pertaining to the pane which are generated by FEEDER.

b. Attitude data for the half-block, but only if the attitude processing option has been specified by the calling program.

c. Digital subcom data, in two forms, and flags pertaining to each word. In one form, the x-ray counts are the source values given on the Decomm tape. In the other form, x-ray counts are corrected for scaling. Digital subcom data is supplied only if the digital subcom processing option has been specified by the calling program.

d. Analog subcom data words and flags for each word. The two words containing command status and housekeeping information are always given. The remaining 8 words are supplied only if the analog subcom processing option has been specified by the calling program.

e. A useful form of the contents of the 1st, 2nd, and 60th of the 64 main frame experiment word fields in the half-block.

f. Counts of data words flagged. The only words counted are those processed by the options in effect.

g. Day-night status as indicated by the main frame flag fields on the Decomm data blocks.

4. The SCAN window

COMMON/SCANS/list

This window is used by FEEDER, when requested, to supply a calling program with main frame data vectors along with corresponding flag vectors. If the calling program had specified complete main frame data during the

initial call to FEEDER, FEEDER assumes the SCAN window to consist of 16 128-byte data vectors followed by 16 128-byte flag vectors. However, if partial main frame data had been specified in the initial call, FEEDER assumes the SCAN window to consist of 16 8-byte data vectors, 16 8-byte flag vectors, and a 3840-byte unused area.

Additional details are given in Section IV.D of this report.

## 5. The INTERRUPTION Vector

COMMON/INTRUP/list

This window contains a collection of flags and other information which indicate certain anomalies in the data stream have been detected by FEEDER.

## C. Summary of commands that a calling program can issue to subprogram FEEDER

1. The initial command. As the name implies, this must be the first command issued and it must be issued once only. The calling program must pass information to be read during the current job step and the options FEEDER is to employ in processing the data.

When FEEDER returns control, it will have read enough of the first file of the first tape to place decision data from the first 16 half-blocks of the file in DECISION window panes 15-30.

## 2. The "advance-DECISION-window-N-panes" command

Normally information for the N oldest half-blocks is lost and the window is filled from the next 32 half-blocks. In other words, if half-blocks  $M-(M+29)$  were in the window before the advance, half-blocks  $(M+N)-(M+N+29)$  will be in the window after a normal advance. The main frame data that FEEDER can supply (in response to a subsequent request for main frame data) is also normally advanced by N half-blocks. Irregular advances are discussed in Section V.B of this report.

It is important to remember that the information in the DECISION window is a copy of corresponding information stored in FEEDER. Thus the calling program could, prior to commanding an advance of the window,



make alterations in the (N+1)th pane. The original contents of this pane would appear in the first pane after FEEDER has executed the advance.

### 3. The "fetch-ORBIT-data-for-time-T" command

If available, the desired information is moved to the ORBIT window. The success or failure of FEEDER'S response is posted in the INTERRUPTION vector.

### 4. The "fetch-file-information" commands

There are 2 commands which may be used, one for the most recently started Decomm file and one for the preceding Decomm file. Information for the appropriate file is moved to the HEADER window.

### 5. The "update-DECISION-window" command

This command allows a calling program to transfer edited DECISION window panes to FEEDER's copy of the DECISION window. FEEDER will then review its revised DECISION window and appropriately reset flags. The edited panes will be shifted by subsequent commands to advance the DECISION window.

### 6. The "update-main-frame-data" command

This command allows a calling program to transfer edited main frame data to FEEDER's copy of this data. The flagged words in the edited version are counted and the count is placed in the corresponding DECISION window pane for the main frame.

### 7. The "delete-half-blocks" command

This command orders FEEDER to "cut and splice" the data stream appearing in the DECISION window. Panes to the left of the specified panes are held in place, the specified panes are lost, panes to the right of those specified are shifted to about those held in place, and new half-blocks are read to complete the DECISION window.

### 8. The "skip-N-file" command

For this activity, the Decomm tape file from which the 30th pane of the DECISION window originated is counted as File 0. The Decomm tape is advanced until the first half-block from file N is in pane 15 of the DECISION window. The last 14 half-blocks of the preceding file will be in panes 1-14 of the DECISION window.

This command can also be used to go directly to the first file of the next tape. In this case no data from the last file of the current tape will be available in panes 1-14.

9. The "fetch-main-frame-data" command

Main frame data corresponding to N consecutive panes of the DECISION window are moved to N consecutive data vectors in the SCAN window, when  $1 < N \leq 16$ . For each main frame data word so moved, its flag field is moved into the corresponding element of the corresponding flag vector. The correspondence of DECISION pane numbers to SCAN window vectors is left solely to the discretion of the calling program.

### III. Suggested Variable Names for Parameters Supplied by Subprogram FEEDER

The variable names listed below were invented primarily to facilitate the writing of this report by providing labels for the parameters whose definitions are given in Section IV. The parameters supplied by FEEDER are defined in the next section, one window at a time. For each window, the parameters are presented in alphabetical order of suggested variable names. To locate a definition, the list on the preceding page shows each of the suggested variable names in alphabetical order and associates each name to the common block in which the variable appears.

VARBL.	COMMON	VARBL.	COMMON	VARBL.	COMMON
ALTKM	ORBIT	HXFS2	INTRUP	QOTIMG	INTRUP
ANXING	ORBIT	HXIOE	INTRUP	QPANY	DECIDE
ATTDAT	DECIDE	HXTIMG	INTRUP	QPATTF	DECIDE
BGAUSS	ORBIT	IHAFN	HEADER	QPFILL	DECIDE
DECMAG	ORBIT	IHANC1	HEADER	QPFILS	DECIDE
ERMCIL	ORBIT	IHANC2	HEADER	QPIOE	DECIDE
GEOLAT	ORBIT	IHANC3	HEADER	QPNITE	DECIDE
GEOLON	ORBIT	IHATTC	HEADER	QPSTAT	DECIDE
HDECIS	DECIDE B	IHBEGT	HEADER	QPTAPS	DECIDE
HHDAYB	HEADER	IHBFN	HEADER	QPTIMG	DECIDE
HHDAYE	HEADER	IHDECN	HEADER	QPTWIL	DECIDE
HHFILN	HEADER	IHDOY	HEADER	QPVAC	DECIDE
HHHRB	HEADER	IHENDT	HEADER	QSANY	INTRUP
HHHRE	HEADER	IHENTN	HEADER	QSFILL	INTRUP
HHMINB	HEADER	IHEXPB	HEADER	QSFILS	INTRUP
HHMINE	HEADER	IHEXTN	HEADER	QSIOE	INTRUP
HHMONB	HEADER	IHMTF	HEADER	QSTAPS	INTRUP
HHMONE	HEADER	IHMTN	HEADER	QSTIMG	INTRUP
HHSECB	HEADER	IHORBC	HEADER	QSVAC	INTRUP
HHSECE	HEADER	IHRN	HEADER	QTAPEN	INTRUP
HHTAPN	HEADER	IHSN	HEADER	QUIT	INTRUP
HHYRB	HEADER	IHSTN	HEADER	QWANY	INTRUP
HHYRE	HEADER	IHYEAR	HEADER	QWFILL	INTRUP
HIDIOE	INTRUP	ITEMID	ORBIT	QWFILS	INTRUP
HIOECT	INTRUP	QDANY	INTRUP	QWIOE	INTRUP
HIXXX1	INTRUP F	QDECIS	DECIDE B	QWTAPS	INTRUP
HNIOE	INTRUP	QDFILL	INTRUP	QWTIMG	INTRUP
HNTIMG	INTRUP	QDFILS	INTRUP	QWVAC	INTRUP
HPAFN	DECIDE	QDIOE	INTRUP	QXSFIL	INTRUP
HPASF	DECIDE	QDTAPS	INTRUP	RAMAG	ORBIT
HPASSV	DECIDE	QDTIMG	INTRUP	SUNXAU	ORBIT
HPBLKN	DECIDE	QDVAC	INTRUP	SUNYAU	ORBIT
HPDFN	DECIDE	QFIOEF	INTRUP	SUNZAU	ORBIT
HPDSF	DECIDE	QHDERR	INTRUP	UDECIS	DECIDE B
HPDSIV	DECIDE	QHENT	HEADER	UNDAWN	HEADER
HPDSSV	DECIDE	QHEXIT	HEADER	UNDAY1	HEADER
HPDTHF	DECIDE	QHIOE	HEADER	UNDAY2	HEADER
HPDTHW	DECIDE	QHMSGG	HEADER	UNDUSK	HEADER
HPDTLW	DECIDE	QHVAC	HEADER	UHLRFP	HEADER
HPFILN	DECIDE	QHXXXX	HEADER F	UHSCID	HEADER
HPIDF	DECIDE	QIXXX1	INTRUP F	UHTANC	HEADER
HPMFN	DECIDE	QIXXX2	INTRUP F	UHTAN2	HEADER
HPSCTS	DECIDE	QIXXX3	INTRUP F	UHTVOL	HEADER
HPSSUM	DECIDE	QIXXX4	INTRUP F	UIXXX1	INTRUP F
HPSYNF	DECIDE	QOANY	INTRUP	UNUSED	HEADER F
HPSYNW	DECIDE	QOCENT	INTRUP	UPTIME	DECIDE
HPTAPN	DECIDE	QOCHNG	INTRUP	UREQT	ORBIT
HPTWIL	DECIDE	QOEOF	INTRUP	UTIME	ORBIT
HSCAN	SCANS B	QOFILS	ORBIT	XKM	ORBIT
HSDATL	SCANS	QOIOE	ORBIT	XKMPS	ORBIT
HSDATS	SCANS	QOIOEF	INTRUP	YKM	ORBIT
HSFLGL	SCANS	QOMISS	INTRUP	YKMPS	ORBIT
HSFLGS	SCANS	QONONE	INTRUP	ZKM	ORBIT
HXFS1	INTRUP	QOTGAP	ORBIT	ZKMPS	ORBIT

'F' DENOTES FILLER VARIABLE. 'B' DENOTES BACKGROUND VARIABLE.

## IV. A. Description of the HEADER window

### 1. Format

The organization of information supplied via the HEADER window is shown in each of Figures IV.A.1, IV.A.2, and IV.A.3. Figure IV.A.1 is a chart of the window and gives a brief description of each parameter. Figure IV.A.2 lists suggested variable names on a similar chart of the window. Figure IV.A.3 shows the Fortran statement to define the suggested variable names.

For all 5 windows, the HEADER window contains the only double word parameters which do not contain a floating point number. The parameters are identified by the variable names UHSCID and UHTVOL.

### 2. Parameter definitions for the HEADER window (in alphabetical order of suggested variable names)

HHDAYB	The day count of month in which the file timespan begins, based on information in the Decomm file label block.
HHDAYE	The day count of month in which the file timespan ends, based on information in the Decomm file label block. The value of HHDAYE will differ from that of HHDAYB if and only if the label block contains an end time of file that is less than the begin time of file.
HHFILN	The sequence count (starting at 1) of the Decomm file within the Decomm tape.
HHHRB	The hours of the begin day that elapse before the begin time of file, based on Decomm file label block information.
HHHRE	The hours of the end day that elapse before the end time of file, based on Decomm file label block information.
HHMINB	Minutes of the begin hour that elapse before the begin time of file, based on Decomm file label block information.
HHMINE	Minutes of the end hour that elapse before the end time of file, based on Decomm file label block information.

HHMONB Month count of year (1 implies January) in which the file timespan begins, based on Decomm file label block information.

HHMONE Month count of year in which the file timespan ends, based on Decomm file label block information.

HHSECB Seconds after the begin minute that the file timespan begins, based on Decomm file label block information.

HHSECE Seconds after the end minute that the file timespan ends, based on Decomm file label block information.

HHTAPN The count of the Decomm tape within the sequence of tapes read during the current job step.

HHYRB The year (4 digits) the file timespan begins, based on Decomm file label block information.

HHYRE The year (4 digits) the file timespan ends, based on Decomm file label block information.

IHAFFN The analog file number given in the file label block.

IHANCL If meaningful, this parameter contains the ascending node count from the first orbit data item supplied for the Decomm file. IHANCL is meaningful only if IHORBC  $\geq$  0.

IHANCL2 If meaningful, this parameter contains the ascending node count from the first "ascending-node-crossing-data-item" in the orbit data supplied for the Decomm file. This parameter is meaningful only if IHORBC  $\geq$  1.

IHANCL3 If meaningful, this parameter contains the ascending node count from the second "ascending -node-crossing-data-item" in the orbit data supplied for the Decomm file. This parameter is meaningful only if IHORBC = 2.

IHATTC The type of attitude computations given in the file label block. (See Attachment I for details.)

IHBEGT The begin time of file (seconds of day) given in the file label block.

IHBFFN The buffer file number given in the file label block.

IHDECN Decomm number given in the file label block.

IHDOY Day count of year from the file label block.

IHENDT End time of file (seconds of day) given in the file label block.

IHENTN      Number of sunlight-entrance times stored in the UNDAWN array.  
 IHEXPN      Experimenter number given in the file label block.  
 IHEXTN      Number of sunlight exit times stored in the UHDUSK array.  
 IHMTF      Master tape file number given in the file label block.  
 IHMTN      Master tape number given in the file label block.  
 IHORBC      IHORBC = -1 implies no ascending node count information is available for the current file. Otherwise IHORBC contains the number of changes in the ascending node count that appear in the orbit data items. (0, 1, or 2 are the only possibilities FEEDER will process.)  
 IHRN      Reel number given in the file label block.  
 IHSN      Station number given in the file label block.  
 IHSTN      Station tape number given in the file label block.  
 IHYEAR      Year (2 digits) given in the file label block.  
 QHENT(4)    QHENT(N) =.TRUE. if and only if UHDAWN(N) was taken from an orbit data block on which a permanent I/O error occurred.  
 QHEXIT(4)   QHEXIT(N) =.TRUE. if and only if UHDUSK(N) was taken from an orbit data block on which a permanent I/O error occurred.  
 QHIOE      QHIOE =.TRUE. if and only if and I/O error occurred on the file label block.  
 QHMSGGS    QHMSGGS =.TRUE. if and only if FEEDER detected 1 or more of the following conditions in the array of information displayed in the HEADER window.  
             1. Contents of UHSCID did not match the value expected by FEEDER.  
             2. Year in label block was not in the 1970's.  
             3. Experimenter number was not 3.  
             4. Duration of file was greater than the maximum expected by FEEDER.

QHVAC        QHVAC =.TRUE. if and only if the contents of the  
 HEADER window contain no meaningful information.

UHDAWN(4)    Array of sunlight-entrance times (seconds of century)  
 taken from the orbit data records. The number of  
 times stored in the array is given by IHENTN and is  
 not expected to exceed 1.

UHDAY1       Time (seconds of century) at 0000 hours of the day  
 the file timespan begins, based on Decomm file label  
 block information. This time is used in computing  
 all other times supplied by FEEDER for data in the  
 file. This time could be helpful to a calling pro-  
 gram in converting times given in seconds of century  
 to seconds of day.

UHDAY2       Time (seconds of century) at 2400 hours of the day  
 the file timespan begins, based on Decomm file label  
 block information.

UHDUSK(4)    Array of sunlight-exit times (seconds of century)  
 taken from the orbit data records. The number of  
 exit times stored in the array is given by IHEXTN  
 and is not expected to exceed 1.

UHLRFP(30)   Array of floating point values given in the file  
 label block. UHLRFP(N) contains the value given in  
 characters (55+6N) through (60+6N) of the label block.

UHSCID       Spacecraft identification from the file label block.  
 This parameter contains character information which  
 can be printed using an A5 Fortran format code.  
 (The 3 low-order bytes of this word contain blanks.)

UHTANC       If meaningful, this parameter contains the time  
 (seconds of century) of the first "ascending-node-  
 crossing-data-item" supplied for the Decomm file.  
 This parameter is meaningful only if IHORBC = 1.

UHTAN2       If meaningful, this parameter contains the time  
 (seconds of century) of the second "ascending-node-  
 crossing-data-item" supplied for the Decomm file.  
 This parameter is meaningful only if IHROBC = 2.

UHTVOL       Decomm tape volume name taken from the DD statement.  
 This parameter contains character information which  
 can be printed using an A6 Fortran format code.  
 (The 2 low-order bytes contain blanks.)



**Figure IV A.1 Parameters Contained in the Header Window**

BYTES	SATELLITE IDENTIFICATION NUMBER (BCD)						
1-8							
9-16	YEAR		STATION NUMBER				
17-24	STATION TAPE NUMBER		ANALOG FILE NUMBER			VALUES FROM BCD PORTION OF LABEL RECORD	
25-32	BUFFER FILE NUMBER		DAY COUNT OF YEAR				
33-40	START TIME OF DAY (SECONDS)		END TIME OF DAY (SECONDS)				
41-48	EXPERIMENTER NUMBER		DECOMM NUMBER				
49-56	REEL NUMBER		MASTER TAPE FILE				
57-64	MASTER TAPE NUMBER		TYPE OF ATTITUDE COMPUTATION				
65-72	NUMBER OF SUNLIGHT ENTRANCE TIMES		I/O ERROR ON FIRST SUN ENT.	I/O ERROR ON SECOND SUN ENT.	I/O ERROR ON THIRD SUN ENT.	I/O ERROR ON FOURTH SUN ENT.	
73-80	FIRST SUN ENTRANCE	TIME (SECONDS OF CENTURY)					SUNLIGHT ENTRANCE AND EXIT TIMES
81-88	SECOND SUN ENTRANCE	TIME (SECONDS OF CENTURY)					
89-96	THIRD SUN ENTRANCE	TIME (SECONDS OF CENTURY)					
97-104	FOURTH SUN ENTRANCE	TIME (SECONDS OF CENTURY)					
105-112	NUMBER OF SUNLIGHT EXIT TIMES		I/O ERROR ON FIRST SUN EXIT	I/O ERROR ON SECOND SUN EXIT	I/O ERROR ON THIRD SUN EXIT	I/O ERROR ON FOURTH SUN EXIT	ENTRANCE EXIT
113-120	FIRST SUNLIGHT EXIT	TIME (SECONDS OF CENTURY)					FROM ORBIT DATA
121-128	SECOND SUNLIGHT EXIT	TIME (SECONDS OF CENTURY)					
129-136	THIRD SUNLIGHT EXIT	TIME (SECONDS OF CENTURY)					
137-144	FOURTH SUNLIGHT EXIT	TIME (SECONDS OF CENTURY)					
145-152	TIME AT 0000 HOURS OF DAY FILE		STARTS (SECONDS OF CENTURY)				TIME CONVERSION
153-160	TIME AT 2400 HOURS OF DAY FILE		STARTS (SECONDS OF CENTURY)				CONSTANTS
161-168	FIRST FLOATING POINT WORD FROM FILE LABEL RECORD						VALUES FROM FLOATING POINT PORTION OF LABEL RECORD
393-400	LAST (30TH) FLOATING POINT WORD FROM FILE LABEL RECORD						
401-408	DECOMM TAPE NAME (FROM NO STATEMENT)						
409-416	TAPE COUNT FOR THIS PROGRAM EXECUTION	FILE COUNT	THIS ARRAY IS NOT IN USE	ERROR MESSAGE FROM HEADREC	I/O ERROR ON LABEL RECORD		HEADER WINDOW IDENTIFIERS
417-424	YEAR(4-DIGITS AT START OF FILE	MONTH COUNT OF YEAR AT START OF FILE	DAY COUNT OF MONTH AT START OF FILE		COMPLETE HOURS OF DAY BEFORE START OF FILE		CALENDAR FORM OF BEGIN TIME AND END TIME
425-432	COMPLETE MINUTES OF HOUR BEFORE START OF FILE	SECONDS OF MINUTE BEFORE START OF FILE	YEAR (4-DIGITS) AT END OF FILE		MONTH COUNT OF YEAR AT END OF FILE		
433-440	DAY COUNT OF MONTH AT END OF FILE	COMPLETE HOURS OF DAY BEFORE END OF FILE	COMPLETE MINUTES OF HOUR BEFORE END OF FILE		SECONDS OF MINUTE BEFORE END OF FILE		
441-448	ASC. NODE DATA INDICATOR		ASC. NODE COUNT AT FILE START			ASCENDING NODE IDENTIFIERS	
449-456	ASC. NODE COUNT AT FIRST CROSSING		ASC. NODE COUNT AT SECOND CROSSING				
457-464	TIME (SECONDS OF CENTURY)	OF FIRST ASC. NODE					
465-472	TIME (SECONDS OF CENTURY)	OF SECOND ASC. NODE CROSSING					
473-480							

BYTES

**Figure IV.A.2 Suggested Variable Names For The Header Window**

1-8	UHSCID							VALUES FROM BCD PORTION OF LABEL RECORD
9-16	IHYEAR			IHSN				
17-24	IHSTN			IHAFN				
25-32	IHBFN			IHDOY				
33-40	IHBEGT			IHENDT				
41-48	IHEXPN			IHDECN				
49-56	IHRN			IHMTF				
57-64	IHMTN			IHATTC				
65-72	IHENTN			QHENT(1)	QHENT(2)	QHENT(3)	QHENT(4)	SUNLIGHT ENTRANCE & EXIT ENTRANCE TIMES EXIT FROM ORBIT DATA RECORDS
73-80	UHDAWN(1)							
81-88	UHDAWN(2)							
89-96	UHDAWN(3)							
97-104	UHDAWN(4)							
105-112	IHEXTN			QHEXIT(1)	QHEXIT(2)	QHEXIT(3)	QHEXIT(4)	
113-120	UHDUSK(1)							
121-128	UHDUSK(2)							
129-136	UHDUSK(3)							
137-144	UHDUSK(4)							
145-152	UHDAY1							TIME CONVERSION CONSTANTS
153-160	UHDAY2							
161-168	UHLRFP(1)							VALUES FROM FLOATING POINT PORTION OF LABEL RECORD
393-400	UHLRFP(30)							
401-408	UHTVOL							
409-416	HHTAPN	HHFILN	QHVAC	QHMSG5	QHIOE			HEADER WINDOW IDENTIFIERS
417-424	HHYRB	HHMONB	HHDAYB		HHHRB			CALENDAR FORM OF BEGIN TIME AND END TIME
425-532	HHMINB	HHSECB	HHYRE		HHMONE			
433-440	HHDAYE	HHHRE	HHMINE		HHSECE			
441-448	IHORBC			IHANC1				ASCENDING NODE IDENTIFIERS
449-546	IHANC2			IHANC3				
547-464	UHTANC							
465-472	UHTAN2							
473-480								

PROGRAM							PUNCHING INSTRUCTIONS	GRAPHIC								
PROGRAMMER						DATE		PUNCH								

[illegible]

## B. Description of the ORBIT window

### 1. Format

The organization of the ORBIT window is shown in Figure IV.B.1 and is briefly described here. Before requesting orbit data, a calling program must enter a request time in the first word (suggested parameter name = UREQT) of the window. FEEDER responds, where possible, by completing the contents of the window. The sequence of parameters in the window is as follows:

- a. Request time (seconds of century).
- b. A vector of 4 data item times (seconds of century) from 4 consecutive data items. Where possible, FEEDER will select data item times so that the request time falls between the second and third of the 4 times.
- c. Seventeen 4-element vectors containing single-precision floating point values. Each vector contains values corresponding to the 4 data-time times for one of the orbit parameters. The vectors are in the same order as listed in Attachment V, with the first vector containing the x-component of the satellite position and the last vector containing the ascending-node-crossing number.
- d. A 4-element full word integer vector containing the type of data item indicators for the 4 data items.
- e. Three 4-element single-byte logical vectors generated by FEEDER. These vectors contain flags pertaining to the data items returned in the window. A summary of these flag settings is posted in the INTERRUPTION vector whenever a calling program requests orbit data.

### 2. Definitions of ORBIT window parameters, by alphabetical order of suggested variable names

- ALTKM(4) ALTKM(N) is the height (km) of the satellite above the spheroid at time = UTIME(N).
- ANXING(4) ANXING(N) is the ascending-node-crossing number at time = UTIME(N).
- BGAUSS(4) BGAUSS(N) is the magnetic field strength (gauss) at time = UTIME(N).

DECMAG(4) DECMAG(N) is the declination (deg) of the magnetic field vector at time = UTIME(N).

ERMCIL(4) ERMIL(N) is the McIlwain L parameter (earth radii) at time = UTIME(N).

GEOLAT(4) GEOLAT(N) is the geodetic latitude (deg) at time = UTIME(N).

GEOLON(4) GEOLON(N) is the geodetic longitude (deg) at time = UTIME(N).

ITEMID(4) ITEMID(N) is the type of data item code for time = UTIME(N).

- 1 implies regular item.
- 2 implies ascending node crossing.
- 3 implies north point item.
- 4 implies descending node crossing.
- 5 implies south point item.
- 6 implies sunlight-entrance item.
- 7 implies sunlight-exit item.

QOFILS(4) QOFILS(N) = .TRUE. if and only if the data item for time = UTIME(N) is the first data item (accepted by FEEDER) for a Decomm file.

QOIOE(4) QOIOE(N) = .TRUE. if and only if a permanent I/O error occurred on the orbit data block containing the data item for time = UTIME(N).

QOTGAP(4) QOTGAP(N) = .TRUE. if and only if the value of UTIME(N) exceeds the value of UTIME(N-1) by more than 60.001 seconds.

RAMAG(4) RAMAG(N) is the right ascension (deg) of the magnetic field vector at time = UTIME(N).

SUNXAU(4) SUNXAU(N) is the x-component (AU) of the solar vector at time = UTIME(N).

SUNYAU(4) SUNYAU(N) is the y-component (AU) of the solar vector at time = UTIME(N).

SUNZAU(4) SUNZAU(N) is the z-component (AU) of the solar vector at time = UTIME(N).

XKM(4) XKM(N) is the x-component (km) of the satellite position vector at time = UTIME(N).

XKMPS(4) XKMPS(N) is the x-component (km/sec) of the satellite velocity vector at time = UTIME(N).

YKM(4)      YKM(N) is the y-component (km) of the satellite position vector at time = UTIME(N).

YKMPS(4)    YKMPS(N) is the y-component (km/sec) of the satellite velocity vector at time = UTIME(N).

ZKM(4)      ZKM(N) is the z-component (km) of the satellite position vector at time = UTIME(N).

ZKMPS(4)    ZKMPS(N) is the z-component (km/sec) of the satellite velocity vector at time = UTIME(N).

UREQT      Time (seconds of century) for which the calling program requested orbit data.

UTIME(4)    Vector of times (seconds of century) for which FEEDER has supplied data items. For N = 1, 2, 3, UTIME (N+1) is always greater than UTIME(N).

PROGRAM	PUNCHING INSTRUCTIONS	GRAPHIC							
PROGRAMMER		PUNCH							
DATE									

C	STATEMENT NUMBER	CONT	FORTRAN STATEMENT																																																															
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
C			IMPLICIT INTEGER*2(H), LOGICAL*1(Q), REAL*8(U)																																																															
C																																																																		
C																																																																		
			COMMON /ORBIT/UREQT, UTIME(4),																																																															
			1 XKM(4), YKM(4), ZKM(4), XKMPS(4), YKMPS(4), ZKMPS(4),																																																															
			1 GEOLON(4), GEOLAT(4), ALTKM(4), SUNXAU(4), SUNYAU(4), SUNZAU(4),																																																															
			1 ERMCIL(4), BGAUSS(4), RAMAG(4), DECMAG(4), ANXING(4),																																																															
			1 ITEMID(4), QOIOE(4), QOFILS(4), QOTGAP(4)																																																															
C			FIGURE IV.B.1. ORBIT WINDOW ORGANIZATION.																																																															

## C. Description of the DECISION window

### 1. Format

The organization of the DECISION window is given in Figure IV.C.3. Figures IV.C.1 and IV.C.2 describe the organization of one pane of the window. Figure IV.C.1 is a chart of a window pane which briefly describes the information contained in each parameter. Figure IV.C.2 is also a chart of a window pane but which shows the assignment of suggested variable names to the parameters.

The variable names suggested for DECISION window parameters were invented merely to aid in defining the organization of the window. Designers of calling programs will likely wish to rename these parameters since the double subscripting of the suggested variables will slow down program execution.

### 2. Definitions of parameters supplied by FEEDER in the DECISION window

These definitions are given in alphabetical order by the suggested variable names. Throughout these definitions, the subscript "N" will indicate pane N of the of the window.

**ATTDAT** ATTDAT(K,N) contains the single-precision form of the K'th attitude word for the half-block. (For further details, see Figures IV.C.1, IV.C.2 and Data Format for OSO-H Experimenter Tape.) If the default option to omit processing attitude data is taken, FEEDER will initialize this array to zero and will not modify the contents of the array during program execution.

**HPAFN** The number of analog subcom words for which some flag is set. In other words, HPAFN(1,N) contains a count of the number of nonzero entries in HPASF(K,N) for  $K = 1, 10$ .

(Programming note: if HPAFN(1,N) = 0, any examination of analog subcom flag settings can be omitted.

**HPASF** HPASF(K,N) is a collection of flags that pertain to the analog subcom word in HPASSV(K,N). If the bits of the half-word HPASF(K,M) are designated by "0000000ihgfdcba", where 0's represent unused bits (set to zero) and "a" is the least significant bit, then:

bit i = 1 if and only if the sync status of the preceding half-block is uncertain because the wheel sync word was either missing or contained fill data.



bit h = 1 if and only if a slip in the wheel subcom data is indicated for the half-block.  
 bit g = 1 if and only if a slip in the wheel subcom data is indicated for the preceding half-block.  
 bit f = 1 if and only if the slip status in the wheel subcom is not clearly indicated for the preceding half-block.  
 bit e = 1 if and only if one or more sync errors occurred for the main frame containing the subcom word.  
 bit d = 1 if and only if one or more sync errors occurred for the main frame preceding the main frame containing the subcom word.  
 bit c = 1 if and only if the sync status of the preceding main frame is uncertain.  
 bit b = 1 if and only if the main frame containing the subcom word is a fill frame.  
 bit a = 1 if and only if fill data is supplied on the Decomm tape for the subcom word.

HPASSV      HPASSV(K,N) contains the value of the K'th analog subcom word for the half-block. By coincidence, HPASSV(K,N) is read out in main frame K and is identified as ASSC word K in the report OSO-H Experiment, Description and Operation, UNH Gamma Monitor, June 2, 1970. The report mentioned above and Figures IV.C.1 and IV.C.2 should adequately define the meaning of each of the 10 subcom words.

HPBLKN      HPBLKN(1,N) is the count of the block within the Decomm file from which the half-block originated.

HPDFN      The number of digital subcom words in the half-block for which some flag is set. In other words, HPDFN(1,N) is the count of nonzero entries in HPDSF(K,N), for K = 1,7.

(Programming note: if HPDFN(1,N) = 0, any examination of digital subcom flags can be omitted.)

HPDSF      HPDSF(K,N) is a collection of flags that pertain to the digital subcom word in both HPDSSV(K,N) and in HPDSIV(K,N). If the half-word HPDSF(K,N) is represented by "0000000ihgfdcba" where 0's represent unused bits (set to zero) and "a" is the least significant bit, then:

bit i = 1 if and only if the digital subcom status for the preceding half-block is uncertain because of either fill data or missing data for the digital subcom sync word.

bit h = 1 if and only if a slip in the digital subcom data is indicated for the half-block.

bit g = 1 if and only if a slip in the digital subcom data is indicated for the preceding half-block.  
 bit f = 1 if and only if the slip status for the preceding half-block is uncertain.  
 bit e = 1 if and only if one or more sync errors occurred for the main frame containing the subcom word.  
 bit d = 1 if and only if one or more sync errors occurred for the preceding main frame.  
 bit c = 1 if and only if the sync error status for the preceding main frame is uncertain.  
 bit b = 1 if and only if the main frame containing the subcom word is a fill frame.  
 bit a = 1 if and only if fill data is supplied on the Decomm tape for the subcom word.

HPDSIV (K,N) is the "interpreted" value of the K'th digital subcom word read out in the major frame represented by the half-block. Only the x-ray counts are interpreted by FEEDER; the time words and the energy ID word are left unchanged. If x is an x-ray count given by the Decomm tape, the interpreted values given in HPKSIV(K,N) for K = 1, 4, 6, or 7 are defined by f(x), where

$$f(i) = \begin{array}{ll} i & \text{if } 0 \leq i \leq 95 \\ 4(i-96) + 98, & \text{if } 96 \leq i \leq 127 \\ 16(i-128) + 232, & \text{if } 128 \leq i \leq 159 \\ 64(i-160) + 768, & \text{if } 160 \leq i \leq 191 \\ 256(i-192) + 2912, & \text{if } 192 \leq i \leq \end{array}$$

HPDSIV (1,N) is the functional interpretation of the digital subcom word value read out in main frame 9. Both the energy identification and the quadrant time for this word, if available, are in the preceding major frame.

HPDSIV(2,N) is the value of the digital subcom word read out in main frame 19. If meaningful, this word contains the energy identification for HPDSIV(K,N), where K = 3, 4, 5, 6, or 7 and for HPDSIV(1, N+1), assuming continuous data.

HPDSIV(3,N) is the value of the digital subcom word read out in main frame 22.

HPDSIV(4,N) is the functional interpretation of the digital subcom word value read out in main frame 23.

HPDSIV(5,N) is the value of the digital subcom word read out in main frame 24.

HPDSIV(6,N) is the functional interpretation of the digital subcom word value read out in main frame 29.

HPDSIV(7,N) is the functional interpretation of the digital subcom word value read out in main frame 47.

The meaning of unflagged entries in this array are indicated in Figures IV.C.1 and IV.C.2 and are more completely described in OSO-H Experiment, Description and Operation, UNH Gamma Ray Monitor, June 2, 1970.

HPDSSV      HPDSSV(K,N) is the source value of the K'th digital subcom word read out in the major frame represented by the half-block.

HPDSSV(K,N) = HPDSIV(K,N) for K = 2,3 or 5.  
 $f(\text{HPDSSV}(K,N)) = \text{HPDSIV}(K,N)$  for K = 1,4,6 or 7, where f is the function described above for variable HPDSIV.

HPDTHF      HPDTHF(1,N) is a collection of flags pertaining to the 60th of the 64 main frame experiment words in the half-block and applies to the contents of HPDTHW(1,N). The pattern of flags contained in HPDTHF(1,N) is the same as for all experiment words. If the bits of HPDTHF(1,N) are represented by "00000000000edcba", where the 0's are unused bits (set to zero) and "a" is the least significant bit, then:

- bit e = 1 if and only if one or more sync errors occurred on at least one of the main frames containing a portion of the experiment word.
- bit d = 1 if and only if one or more sync errors occurred on the main frame that preceded the main frame containing the first portion of the experiment word.
- bit c = 1 if and only if uncertainty exists about the status of the sync errors on the main frame preceding the main frame containing the first portion of the experiment word.
- bit b = 1 if and only if any main frame containing a portion of the experiment word was a fill frame
- bit a = 1 if and only if fill data was supplied for either of the two main frame words containing a portion of the experiment word.

**HPDTHW** HPDTHW(1,N) contains 4 times the value supplied in the 60th of the 64 experiment words in this half-block. If pane(N-5) is the start of a readout for a quadrant of a full scan, then the sum of HPDTHW(1,N) and HPDTHW(1,N-5), if both parameters are meaningful and data are continuous, is the quadrant dead time as read out from the experiment's memory. The flags for HPDTHW(1,N) are given in HPDTHF(1,N). If a quadrant scan does not end on pane N, the contents of HPDTHW(1,N) are meaningless.

**HPDTLW** HPDTLW(1,N) contains the intended value of the two lowest order bits of the second of the 64 experiment words in this half-block. If this pane is the start of a readout for a quadrant of a full scan, then the sum of HPDTLW(1, N) and HPDTHW(1, N+5), if both parameters are meaningful and data are continuous, is the quadrant dead time as read out from the experiment's memory. The flags for HPDTLW(1,N) are given in HPIDF(1, N). If a quadrant scan does not begin in pane N, the value in HPDTLW(1,N) is meaningless.

**HPFILN** HPFILN(1,N) contains the sequence count of the file, within the Decomm tape, from which this half-block was read.

**HPIDF** HPIDF(1,N) contains the collection of flags that pertain to the second of the 64 experiment words in this half-block. The meaning of the individual bits of HPIDF(1,N) is the same as described above for HPDTHF(1,N).

**HPMFN** HPMFN(1,N) is a count of main frame experiment words that were processed from the half-block and which had one or more flags assigned. This count applied only to the main frame data which FEEDER can, under the options in effect, supply to the SCAN window. This count specifically excludes HPDTHW(1,N), HPDTLW(1,N), and HPSYNW(1,N).

(Programming note: If HPMFN(1,N) = 0, examination of flags for main frame data can be omitted for the major frame.)

If FEEDER is commanded to replace a half-block of main frame data corresponding to pane N by an edited version prepared by the calling program, then FEEDER counts the flags of the edited array and inserts the count in HPMFN(1,N). HPMFN(1,N) will remain unchanged if FEEDER is commanded to replace a half-block of decision data by an edited version prepared by a calling program.

**HPSCTS** For the half-block in pane N, HPSCTS(1,N) is the count of nonfill minor frames for which 1 or more sync errors were reported in the flag field.

**HPSSUM** HPSSUM(1,N) is the count of sync bits which were in error in all nonfill minor frames for the half-block in pane N.

**HPSYNF** HPSYNF(1,N) is a collection of flags for the first of the 64 main frame experiment words in the half-block and whose value is given in HPSYNW(1,N). The meaning of the individual bits of HPSYNF(1,N) is the same as described above for HPDTHF(1,N).

**HPSYNW** HPSYNW(1,N) contains the value of the first of the 64 main frame experiment words in the half-block. If this pane is the start of a quadrant scan, then HPSYNW(1,N), is meaningful, is the sync word for the scan and should contain 4095. If pane N is not the start of a quadrant scan, then the contents of HPSYNW(1,N) are not helpful.

**HPTAPN** HPTAPN(1,N) is the sequence count of Decomm tapes (within the job step) from which the half-block was read.

**HPTWIL** HPTWIL(1,N) is the number (from 1 to 48) of the main frame at which a change in the setting of the day-night indicator in the main frame flag field is detected. The main frame number is also taken from the flag field. HPTWIL(1,N) will either contain one of the numbers from 1 to 48 or HPTWIL(1,N) = 0 according as QPTWIL(1,N) = .TRUE. or QPTWIL(1,N) = .FALSE.

**QPANY** QPANY(1,N) = QPVAC(1,N).OR.QPIOE(1,N).OR.QPFILS(1,N).OR.QPTAPS(1,N).OR.QPTIMG(1,N).OR.QPFILL(1,N)  
(Programming note: if QPANY(1,N) = .FALSE., no examination of the other parameters is necessary.)

**QPATTF** Attitude flag field bits are represented, in left-to-right order, by QPATTF(K,N) with QPATTF(14,K) corresponding to the least significant bit of the attitude flag field. For further details see Figures IV.C.1 and IV.C.2 along with Attachment IV.  
  
If the default option is taken to nonprocess attitude data, FEEDER will initialize this parameter's location to zero and will not further modify the area.

**QPFILL** QPFILL(1,N) = .TRUE. if and only if the half-block contains only fill data.

**QPFILS** QPFILS(1,N) = .TRUE. if and only if the half-block is the first half-block from a Decomm file.

QPIOE QPIOE(1,N) = .TRUE. if and only if a permanent I/O error occurred on the data block from which the half-block in this pane was taken.

QPNITE QPNITE(1,N) = .TRUE. if and only if the night indicator is set for the first nonfill main frame of the half-block. If QPFILL(1,N) = .TRUE., then QPNITE(1,N) = .FALSE., but not conversely.

QPSSTAT QPSSTAT(K,N) = .TRUE. if and only if a corresponding bit of the second of the 64 main frame experiment words is set to 1. QPSSTAT(1,N) corresponds to the highest order bit while QPSSTAT(10,N) corresponds to the second from the lowest order bit. Values from QPSSTAT(K,N) are set regardless of main frame options in effect.

If pane N is the start of a quadrant scan and the second of the 64 main frame experiment words is meaningful, QPSSTAT(K,N) will contain true-false interpretation of the main frame status bits. When QPSSTAT(K,N) = .TRUE., its meaning is briefly indicated in Figures IV.C.1 and IV.C.2 and is more completely described in the report OSO-H Experiment, Description and Operation, UNH Gamma Ray Monitor, June 2, 1971.

If pane N is not the start of a quadrant scan, QPSSTAT(K,N) contains useless information.

QPTAPS QPTAPS(1,N) = .TRUE. if and only if the half-block is the first half-block of a Decomm tape.

QPTIMG QPTIMG(1,N) = .TRUE. if and only if the difference

$$\text{UPTIME}(1,N) - \text{UPTIME}(1,N-1)$$

varies from 15.36 seconds by more than 1 second. QPTIMG(1,1) is always set to .FALSE.

FEEDER's copy of QPTIMG(1,N) cannot be reset by a calling program except by revising the time words.

QPTWIL QPTWIL(1,N) = .TRUE. if, for 2 different nonfill main frames, the day-night indicator for the main frame flag fields has different settings. [See QPNITE(1,N) and HPTWIL(1,N).]

QPVAC QPVAC(1,N) = .TRUE. if and only if the current pane N of the window is not in use and does not reflect the contents of any half-block. This condition may be expected at the start of the first Decomm tape and at the end of the last Decomm tape.

UPTIME      UPTIME(1,N) contains the time (seconds of century) at the start of the major frame represented by the half-block.

The computation by FEEDER of values supplied in UPTIME(1,N) is described here to indicate possibilities for miscalculation. For each data block, the value of the milliseconds-of-day field is converted to seconds of day regardless of other anomalies, such as I/O errors, detected in the data block. If the result exceeds 5 seconds before the start time of day (which could be misinterpreted), the value in UHDA11 (which could be wrong) is added to give the time for the first half-block. Otherwise, the value in UHDA12 is added to compute the time of the first half-block. The time for the second half-block is computed by adding 15.36 to the time for first half-block.

Figure IV C.1 Parameters in each Pane of Decision Window

BYTES	1-8	ANY OF NEXT 6 FLAGS SET	PANE NOT IN USE	I/O ERROR FOR THIS PANE	FIRST PANE IN DECOMM FILE	FIRST PANE IN DECOMM TAPE	TIME GAP PRECEDES PANE	THIS PANE IS 100% FILL DATA		PANE FLAGS
	9-16	TAPE VOLUME COUNT		TAPE FILE COUNT		BLOCK COUNT IN FILE				DECOMM TAPE IDENTIFIERS
	17-24	TIME THE MAJOR FRAME STARTS (SECONDS OF CENTURY)								TIME
	25-32	SPIN RATE (RAD/SEC)				PITCH ANGLE (RAD.)				ATTITUDE DATA WORDS
	33-40	ROLL ANGLE (RAD.)				ASPECT ANGLE (RAD.)				
	41-48	RIGHT ASCENSION OF SPIN AXIS (RAD.)				DECLINATION OF SPIN AXIS (RAD.)				
	49-56	RIGHT ASCENSION OF ROLL AXIS (RAD.)				DECLINATION OF ROLL AXIS (RAD.)				
	57-64	RIGHT ASCENSION OF ASPECT AXIS (RAD.)				DECLINATION OF ASPECT AXIS (RAD.)				
	65-72	AZIMUTH TIME OF ENCODER				ATTITUDE FLAG FIELD				
	73-80	NO ASPECT POSSIBLE	NO SPIN POSSIBLE	NO PITCH AVAILABLE	--	--	--	ENCODER PULSES > 1	NO ENCODER PULSE	FLAG FIELD BITS
	81-88	PITCH GAS ACTIVATED	DATA OUT OF BOUNDS	DATA OUT OF BOUNDS	DATA OUT OF BOUNDS	NO DATA	NO DATA	UPPER RANGE BACKGROUND X-RAYS (SOURCE VALUE)		
	89-96	ENERGY ID (SOURCE VALUE)		SOLAR QUADRANT TIME (SOURCE VALUE)		LOWER RANGE SOLAR X-RAYS (SOURCE VALUE)		BACKGROUND QUADRANT TIME (SOURCE VALUE)		XRAY COUNTS AS GIVEN
	97-104	LOWER RANGE BACKGROUND X-RAYS (SOURCE VALUE)		UPPER RANGE SOLAR X-RAYS (SOURCE VALUE)		UPPER RANGE BACKGROUND X-RAYS (INTERPRETED VALUE)		ENERGY ID (SOURCE VALUE)		
	105-112	SOLAR QUADRANT TIME (SOURCE)		LOWER RANGE SOLAR X-RAYS (INTERPRETED VALUE)		BACKGROUND QUADRANT TIME (SOURCE VALUE)		LOWER RANGE BACKGROUND X-RAYS (INTERPRETED VALUE)		XRAY COUNTS INTERPRETED
	113-120	UPPER RANGE SOLAR X-RAYS (INTERPRETED VALUE)		FLAGS UPPER RANGE BACKGROUND X-RAYS		FLAGS ENERGY ID		FLAGS SOLAR QUADRANT TIME		FLAGS
	121-128	FLAGS LOWER RANGE SOLAR X-RAYS		FLAGS BACKGROUND QUADRANT TIME		FLAGS LOWER RANGE BACKGROUND X-RAYS		FLAGS UPPER RANGE SOLAR X-RAYS		
	129-136	COMMAND STATUS		HOUSEKEEPING, COMMAND		7.3-9.1 MEV, BACKGROUND		7.3-9.1 MEV, BACKGROUND		ASSC
	137-144	CP SLAB, BACKGROUND		CP CUP, BACKGROUND		7.3-9.1 MEV, SOLAR		7.3-9.1 MEV, SOLAR		DATA WORDS
	145-152	CP SLAB, SOLAR		CP CUP, SOLAR		FLAGS COMMAND STATUS		FLAGS HOUSEKEEPING COMMAND		
	153-160	FLAGS 7.3-9.1 MEV BACKGROUND		FLAGS 7.3-9.1 MEV, BACKGROUND		FLAGS CP SLAB BACKGROUND		FLAGS CP CUP, BACKGROUND		
	161-168	FLAGS 7.3-9.1 MEV SOLAR		FLAGS 7.3-9.1 MEV, SOLAR		FLAGS CP SLAB SOLAR		FLAGS CP CUP SOLAR		FLAGS
	169-176	VALUE OF FIRST MF EXPERIMENT WORD		FLAGS FIRST MF EXP WORD		VALUE OF 2 LOWEST ORDER BITS OF SECOND MF EXP WD		FLAGS SECOND MF EXPERIMENT WORD		SYNC AND DEAD TIME
	177-184	4X VALUE OF 60TH MF EXPERIMENT WORD		FLAGS 60TH MF EXP WORD		SOLAR	NORMAL QUADRANT	NOT FULL SCAN	INTERMEDIATE SCAN	FIELDS
	185-192	ELECTRONIC CALIBRATION ON	RADIATION CALIB ON	DAY	MAGNETOMETER REFERENCE	CP HVPS ON	GAIN = *1 (HIGH)	NUMBER OF ASSC WORDS WITH ANY FLAG		BITS OF STATUS FIELD
	193-200	NUMBER OF DIGITAL SUBCOM WORD WITH ANY FLAG		NUMBER OF DATA WORDS WITH ANY FLAG		NIGHT	SUN ENTRANCE OR EXIT	MAIN FRAME COUNT AT SUN ENTRANCE OR EXIT		
	201-208	COUNT OF NON-FILL MINOR FRAMES WITH SYNC ERRORS		SUM OF BITS IN ERROR IN NONFILL MINOR FRAMES SYNC		DAY OF YEAR THE MAJOR FRAME STARTS				
	209-216	TIME (SECONDS OF DAY) THE MAJOR FRAME STARTS								
	217-224	SCAN MODE INDICATOR		STATUS BITS (EXCEPT SOLAR-BKGND + DEAD TIME)		UNH POWER ON	UNH POWER OFF	12-BIT STATUS WORD		
	225-232									
	233-240									

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR



**Figure IV.C.2. Suggested Variable Names for the Parameters  
in the N'th Pane of the Decision Window**

BYTES

1-8	QPANY(1,N)	QPAC(1,N)	QPIOE(1,N)	QPFILS(1,N)	QPTAPS(1,N)	QPTMG(1,N)	QPFILL(1,N)		PANE FLAGS
9-16	HPTAPN(1,N)		HPFILN(1,N)		HPBLKN(1,N)				DECOMM TAPE IDENTIFIERS
17-24			UPTIME(1,N)						TIME
25-32	ATTDAT(1,N)				ATTDAT(2,N)				ATTITUDE DATA WORDS
33-40	ATTDAT(3,N)				ATTDAT(4,N)				
41-48	ATTDAT(5,N)				ATTDAT(6,N)				
49-56	ATTDAT(7,N)				ATTDAT(8,N)				
57-64	ATTDAT(9,N)				ATTDAT(10,N)				
65-72	ATTDAT(11,N)				ATTDAT(12,N)				
73-80	QPATTF(1,N)	QPATTF(2,N)	QPATTF(3,N)	QPATTF(4,N)	QPATTF(5,N)	QPATTF(6,N)	QPATTF(7,N)	QPATTF(8,N)	FLAG FIELD
81-88	QPATTF(9,N)	QPATTF(10,N)	QPATTF(11,N)	QPATTF(12,N)	QPATTF(13,N)	QPATTF(14,N)	HPDSSV(1,N)	9	BITS
89-96	HPDSSV(2,N)	19	HPDSSV(3,N)	22	HPDSSV(4,N)	23	HPDSSV(5,N)	24	XRAY COUNTS AS GIVEN
97-104	HPDSSV(6,N)	29	HPDSSV(7,N)	47	HPDSIV(1,N)	9	HPDSIV(2,N)	19	XRAY COUNTS INTERPRETED DSM
105-112	HPDSIV(3,N)	22	HPDSIV(4,N)	23	HPDSIV(5,N)	24	HPDSIV(6,N)	29	
113-120	HPDSIV(7,N)	47	HPDSF(1,N)	9	HPDSF(2,N)	19	HPDSF(3,N)	22	
121-128	HPDSF(4,N)	23	HPDSF(5,N)	24	HPDSF(6,N)	29	HPDSF(7,N)	47	
129-136	HPASSV(1,N)	1	HPASSV(2,N)	2	HPASSV(3,N)	3	HPASSV(4,N)	4	ASSC
137-144	HPASSV(5,N)	5	HPASSV(6,N)	6	HPASSV(7,N)	7	HPASSV(8,N)	8	DATA WORDS
145-152	HPASSV(9,N)	9	HPASSV(10,N)	10	HPASF(1,N)	1	HPASF(2,N)	2	FLAGS
153-160	HPASF(3,N)	3	HPASF(4,N)	4	HPASF(5,N)	5	HPASF(6,N)	6	
161-168	HPASF(7,N)	7	HPASF(8,N)	8	HPASF(9,N)	9	HPASF(10,N)	10	
169-176	HPSYNW(1,N)		HPSYNF(1,N)		HPDTLW(1,N)		HPIDF(1,N)		
177-184	HPDTHW(1,N)		HPDTHF(1,N)		QPSTAT(1,N)	QPSTAT(2,N)	QPSTAT(3,N)	QPSTAT(4,N)	SYNC AND DEADTIME FIELDS
185-192	QPSTAT(5,N)	QPSTAT(6,N)	QPSTAT(7,N)	QPSTAT(8,N)	QPSTAT(9,N)	QPSTAT(10,N)	HPAFN(1,N)		BITS OF STATUS FIELD
193-200	HPDFN(1,N)		HPMFN(1,N)		QPNITE(1,N)	QPTWIL(1,N)	HPTWIL(1,N)		
201-208	HPSCTS(1,N)		HPSSUM(1,N)			IPDOY(1,N)			
209-216			UPSOD(1,N)						
217-224	HPMODE(1,N)		HPSTAT(1,N)		QPWON(1,N)	QPWOFF(1,N)	HPSTWD(1,N)		
225-232									
233-240									

[illegible][illegible]



## D. The SCAN data window

### 1. Organization

When requested to supply main frame experiment words to the SCAN data window, FEEDER assumes the window to begin with 16 data vectors followed by 16 flag vectors. (During initialization FEEDER zeros the entire 4096 bytes of the window.) All vectors are assumed by FEEDER to be either 8 bytes in length or 128 bytes in length, depending upon whether partial or complete main frame options, respectively, had been specified during the initial call to FEEDER.

The correspondence of DECISION window pane sequence numbers to vector sequence numbers is controlled in the calling program by parameters passed to FEEDER when a request is made for main frame data to be moved to the window. However, if FEEDER moves data into data vector N, the flag fields for the data are always moved into flag vector N.

### 2. Parameter definitions

HSCAN	The purpose of this parameter is to define the length of the window and to serve as a background parameter. However, when the complete main frame data option is in effect, HSCAN(K,N,1) contains the data value for the K'th experiment word in the N'th data vector while HSCAN(K,N,2) contains the corresponding flag field. The meanings of the bits of HSCAN(K,N,2) are the same as described for HPDTHF in the DECISION window.
HSDATL	HSDATL(K,N) contains the data value for the K'th experiment word stored in the N'th full length data vector. This parameter is to be used only when the complete main frame data option has been specified at initialization.
HSDATS	HSDATS(K,N) contains the value for the (K+60)th experiment word for the half-block whose data values are stored in the N'th short length (8 bytes) data vector. This parameter is to be used only when the partial main frame data option has been specified.
HSFLGL	HSFLGL(K,N) contains the flags pertaining to the data value contained in HSDATL(K,N). The bit meanings of HSFLGL(K,N) are the same as described for HPDTHF in the DECISION window. Like HSDATL, HSFLGL should be used only when the complete main frame data option has been specified.

HSFLGS      HSFLGS(K,N) contains the flags pertaining to the data value contained in HSDATS(K,N). Like HSDATS, HSFLGS should be used only when the partial main frame data option has been specified. The bit meanings of HSFLGS are the same as described for HPDTHF in the DECISION window (see Section IV.C.2).

[illegible]

LINE	STATEMENT NUMBER	CONT.	FORTRAN STATEMENT
C	1		IMPLICIT INTEGER*2(H), LOGICAL*1(Q), REAL*8(U)
C	2		COMMON /SCANS/ HSCAN(64,16,2)
	3		DIMENSION HSDATL(64,16), HSDATS(4,16), HSFLGL(64,16),
	4		HSFLGS(4,16)
	5		EQUIVALENCE (HSCAN(1,1,1), HSDATL(1,1), HSDATS(1,1))
	6		(HSCAN(1,2,1), HSFLGS(1,1)), (HSCAN(1,1,2), HSFLGL(1,1))
C	7		FIGURE IV.D.2. SCAN DATA WINDOW ORGANIZATION.

## E. Description of the INTERRUPTION vector window

### 1. Format

The organization of information supplied to a calling program via the INTERRUPTION vector is shown in each of Figures IV.E.1, IV.E.2, and IV.E.3. Figure IV.E.1 is a chart of the vector which hints at the meanings of the various parameters. Figure IV.E.2 is a similar chart which shows the assignment of suggested variable names to the parameters. Figure IV.E.3 is the Fortran statement necessary to define the suggested variables.

FEEDER employs different guidelines for resetting different portions of the INTERRUPTION vector. One section is always reset; other sections are completely reset, but for certain activities only; still other sections are treated as scratch areas to be used as needed.

a. Bytes 1-8 are always reset. A calling program must save any desired information from this portion before issuing another call to FEEDER.

b. Bytes 9-24 are completely reset whenever a calling program requests orbit data. These bytes are not modified by other commands.

c. The parameters in bytes 25-49, 55-56, 65-66 are a function of the current status of the DECISION window and are reset by FEEDER whenever a command affects the contents of the window.

- (1) "Initial" command.
- (2) "Advance-DECISION-window-N-panes" command.
- (3) "Update-DECISION-window" command.
- (4) "Delete-half-blocks" command.
- (5) "Skip-N-files" command.

d. The remaining parameters of the DECISION window are used only as needed. The meaningfulness of such parameters is always indicated elsewhere.

2. Definitions of parameters in the INTERRUPTION vector, by alphabetical order of suggested variable names.

HIDIOE(8,6) Array of identifiers of Decomm tape blocks which caused permanent I/O errors. (This array is updated only when the occurrence of I/O errors is indicated both by HIOECT  $\neq$  0 and by QFIOEF = .TRUE.) Let N be the lesser of 6 and HIOECT. Then, for J = 1,N,

HIDIOE(1,J) is the Decomm tape count within the current run for which, during last tape reading by FEEDER, the Jth I/O error occurred.

HIDIOE(2,J) is the file count within the tape for the Jth I/O error.

HIDIOE(3,J) is the block count within the file for the Jth I/O error.

HIDIOE(4,J), HIDIOE(5,J), and HIDIOE(6,J) exist merely as a convenience to FEEDER and contain no helpful information.

HIDIOE(7,J) contains the sense information provided in the 3rd and 4th bytes of the IOB (input-output block for the data set) after the I/O error was detected.

HIDIOE(8,J) contains the status information from the channel status word after the I/O error was detected. FEEDER obtains this information from the 13th and 14th bytes of the IOB.

HIOECT The count of permanent I/O errors that occurred during the last call to FEEDER. (This parameter is reset with every call to FEEDER.)

HNIOE The count of panes in the DECISION window which result from Decomm tape blocks causing permanent I/O errors. (This parameter is updated each time the DECISION window is modified.)

HNTIMG The count of panes in the DECISION window for which the time disagrees with the time of the preceding pane. (This parameter is updated every time the DECISION window is modified.)

HXFS1 The number of the first pane, if any, of the DECISION window which results from the first half-block of a Decomm tape file. (This parameter is updated only if QWFILS = .TRUE.)



- HXFS2      The number of the second pane, if any, in the DECISION window which results from the first half-block of a Decomm tape file. (This parameter is updated only if QXSFIL = .TRUE.) All higher-numbered panes must be individually examined for this condition.
- HXIOE(4)    Vector of pane numbers which resulted from data blocks for which permanent I/O errors occurred. The contents of this vector are meaningless unless QWIOE = .TRUE. If meaningful, let N be the lesser of 4 and HNIOE. Then for  $J = 1, N$ , HXIOE(J) is the number of the Jth pane which is flagged as resulting from an I/O error.
- If the value of HXIOE should exceed 4, the calling program will have to fish through the remainder of the DECISION window to locate those panes, after the 4th, which are flagged as resulting from an I/O error.
- HXTIMG(19) Vector of pane numbers which are flagged for time disagreement with preceding panes. (See QPTIMG definition in Section IV.C.2 of this report.) The contents of this vector are meaningful only if a time disagreement exists and is indicated both by QWTIME = .TRUE. and by HNTIME  $\neq$  0. If meaningful, let N be the lesser of the 19 and HNTIME. Then, for  $J = 1, N$ , HXTIMG(J) is the number of the Jth pane to be flagged for time disagreement.
- It seems preposterous to consider the possibility of overflowing this array. However, should HNTIME exceed 19, and the calling program be too stubborn to lose interest, then the remainder of the DECISION window will have to be searched for other time gaps.
- QDANY      "Any-flag" indicator for the first 14 panes of the DECISION window. (This variable is updated whenever the DECISION window is modified.)  
             QDANY=QPANY (1,1).OR.QPANY (1,2).OR. ... .OR.QPANY(1,14)
- QDFILL     "Any-fill-half-block" indicator for the first 14 panes of the DECISION window. (This variable is updated whenever the DECISION window is modified.)  
             QDFILL=QPFILL(1,1).OR.QPFILL(1,2).OR. ... .OR.QPFILL(1,14)
- QDFILS     "Any-first-in-file-half-block" indicator for the first 14 panes of the DECISION window. (This variable is updated whenever the DECISION window is modified.)  
             QDFILS=QPFILS(1,1).OR.QPFILS(1,2).OR. ... .OR.QPFILS(1,14)

**QDIOE** "Any-I/O-error" indicator for the first 14 panes of the DECISION window. (This variable is updated whenever the DECISION window is modified.)  
 QDIOE=QPIOE(1,1).OR.QPIOE(1,2).OR. ... .OR.QPIOE(1,14)

**QDTAPS** "Any-first-in-tape-half-block" indicator for the first 14 panes of the DECISION window. (This variable is updated whenever the DECISION window is modified.)  
 QDTAPS=QPTAPS(1,1).OR.QPTAPS(1,2).OR. ... .OR.QPTAPS(1,14)

**QDTIMG** "Any-time-gap" indicator for the first 14 panes of the DECISION window. (This variable is updated whenever the DECISION window is modified.)  
 QDTIMG=QPTIMG(1,1).OR.QPTIMG(1,2).OR. ... .OR.QPTIMG(1,14)

**QDVAC** "Any-vacancy" indicator for the first 14 panes of the DECISION window. (This variable is updated whenever the DECISION window is modified.)  
 QDVAC=QPVAC(1,1).OR.QPVAC(1,2).OR. ... .OR.QPVAC(1,14)

**QFIOEF** =.TRUE. if and only if a permanent I/O error occurred during the last call to FEEDER. (This variable is reset with every call to FEEDER.)

**QHDERR** =.TRUE. if and only if the last call to FEEDER caused the processing of a file label record containing unacceptable conditions. QHDERR=.TRUE. always causes QUIT to be set to .TRUE., but not conversely. This variable is updated with every call to FEEDER.

**QOANY** "Any-flag" indicator for response to request of orbit data. (This variable is updated with every request for orbit data.)

**QOCENT** "Off-center" indicator for orbit data. QOCENT=.TRUE. implies the request time does not fall between the second and third times of the orbit data items returned. It also implies the request time is near an extreme limit of currently stored orbit data. (This variable is updated with every request for orbit data.)

**QOCHNG** "ORBIT-count-change" indicator. QOCHNG=.TRUE. implies the 4 data items returned do not all have the same ascending node count. (This variable is updated with every request for orbit data.)

**QOEOF** "Any-EOF" indicator for orbit data. QOEOF=.TRUE. implies the orbit data items returned do not all come from the same file of the Decomm tape. (This variable is updated with every request for orbit data.)

- QOIOEF "Any-I/O-error" indicator for orbit data. QOIOEF=.TRUE. implies at least one of the orbit data items returned was taken from a block for which a permanent I/O error occurred. (This variable is updated with every request for orbit data.)
- QOMISS "Unable-to-supply" indicator for orbit data. QOMISS=.TRUE. implies no data was available for the requested time. (This variable is updated with every request for orbit data.)
- QONONE "No-orbit-data available" indicator. QONONE=.TRUE. implies FEEDER has no orbit data items stored. Likely causes of this condition are failure to specify orbit data processing option or lack of orbit data on quick-look tapes. (This variable is updated with every request for orbit data.)
- QOTIMG "Time-gap" indicator for orbit data. QOTIMG = .TRUE. implies that, for a consecutive pair of the 4 data items returned, the difference between the data item times exceeded 60.001 seconds. (This variable is updated with every request for orbit data.)
- QSANY "Any-flag" indicator for the last 16 panes of the DECISION window. (This variable is updated whenever the DECISION window is modified.)  
QSANY=QPANY(1,15).OR.QPANY(1,16).OR. ... .OR.QPANY(1,30)
- QSFILL "Any-fill-half-block" indicator for the last 16 panes of the DECISION window. (This variable is updated whenever the DECISION window is modified.)  
QSFILL=QPFILL(1,15).OR.QPFILL(1,16).OR. ... .OR.QPFILL(1,30)
- QSFILS "Any-first-in-file-half-block" indicator for the last 16 panes of the DECISION window. (This variable is updated whenever the DECISION window is modified.)  
QSFILS=QPFILS(1,15).OR.QPFILS(1,16).OR. ... .OR.QPFILS(1,30)
- QSIOE "Any-I/O-error" indicator for the last 16 panes of the DECISION window. (This variable is updated whenever the DECISION window is modified.)  
QSIOE=QPIOE(1,15).OR.QPIOE(1,16).OR. ... .OR.QPIOE(1,30)
- QSTAPS "Any-first-in-tape-half-block" indicator for the last 16 panes of the DECISION window. (This variable is updated whenever the DECISION window is modified.)  
QSTAPS=QPTAPS(1,15).OR.QPTAPS(1,16).OR. ... .OR.QPTAPS(1,30)

QSTIMG "Any-time-gap" indicator for the last 16 panes of the DECISION window. (This variable is updated whenever the DECISION window is modified.)  
QSTIMG=QPTIMG(1,15).OR.QPTIMG(1,16).OR. ... .OR.QPTIMG(1,30)

QSVAC "Any-vacancy" indicator for the last 16 panes of the DECISION window. (This variable is updated whenever the DECISION window is modified.)  
QSVAC=QPVAC(1,15).OR.QPVAC(1,16).OR. ... .OR.QPVAC(1,30)

QTAPEN =.TRUE. if and only if FEEDER has encountered the end of the last tape volume to be processed. This variable is updated with every call to FEEDER.

QUIT =.TRUE. only if further processing is either impossible or counter-productive. (This variable is updated with every call to FEEDER.)

QWANY "Any-flag" indicator for the entire DECISION window.  
QWANY = QDANY.OR.QSANY

QWFILL "Any-fill-half-block" indicator for the DECISION window.  
QWFILL = QDFILL.OR.QSFILL

QWFILS "Any-first-in-file-half-block" indicator for the DECISION window.  
QWFILS = QDFILS.OR.QSFILS

QWIOE "Any-I/O-error" indicator for the DECISION window.  
QWIOE = QDIOE.OR.QSIOE

QWTAPS "Any-first-in-tape-half-block" indicator for the DECISION window.  
QWTAPS = QDTAPS.OR.QSTAPS

QWTIMG "Any-time-gap" indicator for the DECISION window.  
QWTIMG = QDTIMG.OR.QSTIMG

QWVAC "Any-vacancy" indicator for the DECISION window.  
QWVAC = QDVAC.OR.QSVAC

QXSFIL "Excess-file" indicator for the DECISION window.  
QXSFIL=.TRUE. implies more than one half-block is flagged for being the first half-block from a Decomm tape file. (This variable is updated whenever the DECISION window is modified.)

Figure IV E.1.

## Interruption Vector Parameters

Figure 14.2.1.

Interpretation Vector Parameters

BYTES	1-8	CROSS ANOMALY	I/O ERROR JUST OCCURRED	NUMBER OF I/O ERRORS THAT JUST OCCURRED		HEADER DATA UNACCEPTABLE	END OF LAST TAPE DETECTED			GENERAL
	9-16	ANY OF NEXT 7 OR QUIT	ORBIT DATA OFF CENTER	EOF EMBEDDED IN DATA ITEMS	NO ORBIT ITEMS RETURNED	TIME GAP IN ORBIT ITEMS	I/O ERROR ON AN ORBIT ITEM	ASCENDING NODE COUNT CHANGE	NO ORBIT DATA AVAILABLE	RESPONSE FLAGS WHEN ORBIT DATA REQUESTED
	17-24									
	25-32	ANY PANE FLAG	ANY EMPTY PANES	ANY PANES WITH I/O ERRORS	ANY NEW FILES	ANY NEW TAPES	ANY TIME GAP BETWEEN PANES	ANY PANES WITH 100% FILL		ALL WINDOW PANE FLAG
	33-40	"	"	"	"	"	"	"		
	41-48	"	"	"	"	"	"	"		PANES 1-14 PANES 15-30
	49-56	2 OR MORE NEW FILES		NUMBER OF FIRST PANE FROM FIRST NEW FILE		NUMBER OF FIRST PANE FROM SECOND NEW FILE		COUNT OF PANES WITH I/O ERRORS		NEW FILE INDICATORS
	57-64	NUMBER OF FIRST PANE WITH I/O ERROR		NUMBER OF SECOND PANE WITH I/O ERROR		NUMBER OF THIRD PANE WITH I/O ERROR		NUMBER OF FOURTH PANE WITH I/O ERROR		COUNT,
	65-72	COUNT OF TIME GAPS BETWEEN PANES		NUMBER OF FIRST PANE PRECEDED BY TIME GAP		NUMBER OF SECOND PANE PRECEDED BY TIME GAP		NUMBER OF THIRD PANE PRECEDED BY TIME GAP		INDICES OF I/O ERROR PANES
	73-80									
	81-88									COUNT, INDICES
	89-96									OF TIME GAPS
	97-104	NUMBER OF 16TH PANE PRECEDED BY TIME GAP		NUMBER OF 17TH PANE PRECEDED BY TIME GAP		NUMBER OF 18TH PANE PRECEDED BY TIME GAP		NUMBER OF 19TH PANE PRECEDED BY TIME GAP		IN DECISION
	105-112	DECOMM TAPE COUNT FOR FIRST I/O ERROR		FILE COUNT FOR FIRST I/O ERROR		BLOCK COUNT FOR FIRST I/O ERROR				
	113-120					3RD, 4TH BYTES OF IOB AFTER FIRST I/O ERROR		13TH, 14TH BYTES OF IOB AFTER FIRST I/O ERROR		1ST
	121-128									2ND
	129-136									I/O ERROR IDENTIFIERS,
	137-144									3RD SENSE BYTES,
	145-152									FROM FEEDER's
	153-160									4TH
	161-168									MOST RECENT
	169-176									INPUT ERRORS
	177-184									5TH
	185-192	DECOMM TAPE COUNT FOR SIXTH I/O ERROR		FILE COUNT FOR SIXTH I/O ERROR		BLOCK COUNT FOR SIXTH I/O ERROR				6TH
	193-200					3RD, 4TH BYTES OF IOB AFTER SIXTH I/O ERROR		13TH, 14TH BYTES OF IOB AFTER SIXTH I/O ERROR		

BYTES

Figure IV. E. 2 Suggested Variable Names For Interruption Vector.

1-8	QUIT	QFIOEF	HIOECT	QHDERR	QTAPEN		GENERAL	
9-16	QOANY	QOCENT	QOEOF	QOMISS	QOTIMG	QOIOEF	QOCHNG	QONONE
17-24							RESPONSE FLAGS WHEN ORBIT DATA REQUESTED	
25-32	QWANY	QWVAC	QWIOE	QWFILS	QWTAPS	QWTIMG	QWFILL	
33-40	QDANY	QDVAC	QDIOE	QDFILS	QDTAPS	QDTIMG	QDFILL	
41-48	QSANY	QSVAC	QSIOE	QSFILS	QSTAPS	QSTIMG	QSFILL	
49-56	QXSFIL		HXFS1		HXFS2		HNIOE	
57-64		HXIOE(1)	HXIOE(2)		HXIOE(3)		HXIOE(4)	
65-72		HNTIMG	HXTIMG(1)		HXTIMG(2)		HXTIMG(3)	
73-80		HXTIMG(4)	HXTIMG(5)		HXTIMG(6)		HXTIMG(7)	
81-88		HXTIMG(8)	HXTIMG(9)		HXTIMG(10)		HXTIMG(11)	
89-96		HXTIMG(12)	HXTIMG(13)		HXTIMG(14)		HXTIMG(15)	
97-104		HXTIMG(16)	HXTIMG(17)		HXTIMG(18)		HXTIMG(19)	
105-112		HIDIOE(1,1)	HIDIOE(2,1)		HIDIOE(3,1)			
113-120					HIDIOE(7,1)		HIDIOE(8,1)	
121-128		HIDIOE(1,2)	HIDIOE(2,2)		HIDIOE(3,2)			
129-136					HIDIOE(7,2)		HIDIOE(8,2)	
137-144		HIDIOE(1,3)	HIDIOE(2,3)		HIDIOE(3,3)			
145-152					HIDIOE(7,3)		HIDIOE(8,3)	
153-160		HIDIOE(1,4)	HIDIOE(2,4)		HIDIOE(3,4)			
161-168					HIDIOE(7,4)		HIDIOE(8,4)	
169-176		HIDIOE(1,5)	HIDIOE(2,5)		HIDIOE(3,5)			
177-184					HIDIOE(7,5)		HIDIOE(8,5)	
185-192		HIDIOE(1,6)	HIDIOE(2,6)		HIDIOE(3,6)			
193-200					HIDIOE(7,6)		HIDIOE(8,6)	

ALL WINDOW PANE  
PANES 1-14 FLAG  
PANES 15-30 SUMMARIES  
NEW FILE INDICATORS  
COUNT,  
INDICES OF I/O  
ERROR PANES  
COUNT, INDICES  
OF TIME GAPS  
IN DECISION  
WINDOW.  
1ST  
I/O ERROR  
IDENTIFIERS,  
SENSE BYTES,  
FROM  
FEEDER'S  
MOST RECENT  
INPUT ERRORS  
2ND  
3RD  
4TH  
5TH  
6TH

PROGRAM		PUNCHING INSTRUCTIONS	GRAPHIC																
PROGRAMMER		DATE		PUNCH															

COMM.	STATEMENT NUMBER	CONT.	FORTRAN STATEMENT
	1		IMPLICIT INTEGER*2(H), LOGICAL*1(Q), REAL*8(U)
	2		COMMON /INTRUP/QUIT QFIOEF HIODECT QHDERR QTAPEN HIXXX1
	3		, QOANY, QOCENT, QOEOP, QOMISS, QOTIMG, QOIOEF, QOCHNG, QOONE, QIXXX1,
	4		QWANY, QWVAC, QWIOE, QWFILS, QWTAPS, QWTIMG, QWFILL, QIXXX1,
	5		QDANY, QDVAC, QDIOE, QDFILS, QDTAPS, QDTIMG, QDFILL, QIXXX2,
	6		QSANY, QSVAC, QSTIOE, QSFILS, QSTAPS, QSTIMG, QSFILL, QIXXX3,
	7		QXSFIL, QIXXX4, HXFS1, HXFS2, HNTIOE,
	8		HXTIOE(4), HNTIMG, HXTIMG(19), HIDIODE(8,6)
	9		
	10		
	11		
	12		
	13		
	14		
	15		
	16		
	17		
	18		
	19		
	20		
	21		
	22		
	23		
	24		
	25		
	26		
	27		
	28		
	29		
	30		
	31		
	32		
	33		
	34		
	35		
	36		
	37		
	38		
	39		
	40		
	41		
	42		
	43		
	44		
	45		
	46		
	47		
	48		
	49		
	50		
	51		
	52		
	53		
	54		
	55		
	56		
	57		
	58		
	59		
	60		
	61		
	62		
	63		
	64		
	65		
	66		
	67		

C • FIGURE IV.E.3. SUGGESTED VARIABLES FOR INTERRUPTION VECTOR.

## V. Detailed Description of the Commands to which FEEDER Responds

A sample CALL statement is provided for each command to provide a vehicle for defining the arguments passed. The first argument of every CALL statement identifies the command FEEDER is to execute.

Since the commands are presented in no particular sequence, the following index may be helpful.

<u>Section</u>	<u>Command Purpose</u>
V.A	Initialization.
V.B	Advancing the Decomm tape.
V.C	Retrieval of orbit data.
V.D	Retrieval of file label information.
V.E	Changing the values in the DECISION window.
V.F	Changing the values in the main frame data.
V.G	Deleting portions of the DECISION window.
V.H	File skipping and tape skipping.
V.I	Displaying main frame data.

### A. The "initial" command

```
CALL FEEDER (1,NTAPES,MFDOP,IATTOP,IDSOP,IASOP,IORBOP)
```

NTAPES      The exact number of Decomm tapes to be processed during the job step.

MFDOP      Option code to control the processing of main frame data to be supplied to the SCAN window.

MFDOP = 0 implies main frame data is not to be processed. (The 1st, 2nd, and 60th experiment words are processed for the DECISION window regardless of the option code passed via MFDOP.)

MFDOP = 1 implies all main frame data is to be processed. When subsequently requested to supply main frame data, FEEDER assumes the SCAN window to consist of 16 128-byte data vectors followed by 16 128-byte flag vectors.

MFDOP = 2 implies PHA channel counts are to be ignored. FEEDER will respond by processing the 61st through 64th experiment words for each half-block. (Of course, if the half-block does not end a quadrant scan, the 61st-64th experiment words will again be unwanted channel counts.) Only these 4



experiment words will be tested to establish the flag count contained in the `HPMFN(1,N)` variable in the DECISION window. When subsequently requested to supply main frame data, FEEDER will assume the SCAN data window to consist of 16 8-byte data vectors followed by 16 8-byte flag vectors.

If the value of `MFDOP` is not 0, 1, or 2, FEEDER will set `QUIT = .TRUE.` and assume a value of 0.

**IATTOP**      Option code to control the processing of attitude data words.

`IATTOP = 0` implies omission of attitude data from the DECISION window for the current job step. The `ATTDAT` and `QPATTF` variable locations (see Figure IV.C.2) will be initially zeroed and subsequently ignored by FEEDER.

`IATTOP ≠ 0` implies attitude data values are to be processed and to be placed in the DECISION window.

**IDSOP**      Option code to control the processing of digital subcom data.

`IDSOP = 0` implies omission of digital subcom data from the DECISION window during the current job step. The `HPDSSV`, `HPDSIV`, `HPDSF` variable locations (see Figure IV.C.2) will be initially zeroed and subsequently ignored by FEEDER.

`IDSOP ≠ 0` implies digital subcom data are to be processed and to be placed in the DECISION window.

**IASOP**      Option code to control the processing of analog subcom data words.

`IASOP = 0` implies omission of processing wheel analog subcom words 3-10. Words 1 and 2 are processed regardless of the option specified by `IASOP`. The locations reserved in the DECISION window for words 3-10, and for their associated flags, are initialized to zero and are subsequently ignored by FEEDER. (See variables `HPASSV` and `HPASF` in Figure IV.C.2) Only words 1 and 2 will be examined to determine the count of flagged words for the variable `HPAFN(1,N)` in the DECISION window panes.

`IASOP ≠ 0` implies all 10 wheel analog subcom words are to be processed and to be placed in the DECISION window.

IORBOP      Option code to control the processing of orbit data.

IORBOP = 0 implies no orbit data needed during the current job step. If a subsequent request for orbit data is made, FEEDER will assume the calling program is insane and will set QUIT = .TRUE.

IORBOP = 1 implies only regular orbit data items are desired. FEEDER will store the first data item for each file, regardless of data item type. Subsequent special data items will be discarded and only regular data items stored to supply requests for orbit data.

IORBOP = 2 implies all orbit data items required.

If the value of IORBOP is not 0, 1, or 2. FEEDER will assume a value of 0 and will set QUIT = .TRUE.

Orbit data blocks are scanned for sun-entrance times, for sun-exit times and for ascending node crossings, regardless of the option specified by IORBOP.

#### B. The "advance-DECISION-window-N-panes" command

CALL FEEDER (2,N)

Where the value of N is the number of panes the DECISION window and associated main frame data are to be advanced. The response made by FEEDER varies with the value of N.

N < 0. FEEDER sets QUIT = .TRUE. and takes no further action.

N = 0. The display FEEDER most recently made in the DECISION window is reproduced. No other action takes place.

$1 \leq N \leq 16$ . The contents of the first N panes are lost. The contents of the remaining (30-N) panes are shifted to the first (30-N) panes so that what was formerly in the (N+1)st pane is now in the 1st pane; what was formerly in the (N+2)nd pane is now in the 2nd pane, etc. Then N half-blocks are read from the Decomm tape to complete the new DECISION window display. The set of main frame data stored by FEEDER is updated to correspond to the new contents of panes 15-30 of the DECISION window, provided a main frame processing option has been elected. Appropriate settings are next made for the QPANY(1,N) and the OPTIMG(1,N) flags. Finally, flag summaries and associated pointers are determined and placed in the INTERRUPTION vector.

$N \geq 17$ . FEEDER recognizes this as a plunge forward into the unknown and will limit the advance, if necessary, to guarantee the contents of pane 14 after the advance will have originated from the same file as did pane 30 before the advance. Otherwise, FEEDER's response in the same as for  $1 \leq N \leq 16$ .

C. The "fetch-ORBIT-data-for-time-T" command

CALL FEEDER (3)

The ORBIT window variable UREQT is assumed by FEEDER to contain the request time T.

If the option had been specified to omit orbit data processing, FEEDER will set QUIT = .TRUE., appropriately set response flags in bytes 9-16 of the INTERRUPTION vector, but take no other action. In particular, the ORBIT window is unmodified by the command.

If time T falls within the range of available data, FEEDER will complete the ORBIT window. Let  $t_1, t_2, \dots, t_n$  denote the times for which FEEDER has stored orbit data items. FEEDER will complete the ORBIT window if  $t_1 < T < t_n + 60.000$ , where all times are in seconds. If T falls near enough to the lower (or upper) limit so that  $T < t_2$  (or  $T > t_{n-1}$ ), then FEEDER will enter the first 4 (or last 4) data items in the ORBIT window and will set QOCENT = .TRUE. Ordinarily FEEDER will select 4 consecutive data items for times  $t_i, t_{i+1}, t_{i+2}$ , and  $t_{i+3}$  so that  $t_{i+1} < T \leq t_{i+2}$ .

If time T falls outside the range of available data ( $T < t_1$  or  $T > t_n + 60.000$  or no data available), FEEDER appropriately sets the response flags in bytes 9-16 of the INTERRUPTION vector, but takes no other action. In particular, the ORBIT window remains unchanged by the command.

D. The "fetch-file-information" commands

Two commands are available, one to obtain the information for the newer file currently being read by FEEDER and (older) file. For either command, the retrieved information is placed in the HEADER window.

# 1. Retrieving information from the older file

CALL FEEDER (4)

There is no information for an older file at the start of the first tape. Under certain circumstances there may be no information for an older file at the start of subsequent tapes. Consequently, it would be prudent to follow the command by examining the QHVAC parameters in the HEADER window.

# 2. Retrieving information from the newer file

CALL FEEDER (5)

## E. The "update-DECISION-window" command

This command allows a calling program to edit one or more panes of the DECISION window. Displays made in the DECISION window are copies of a storage area containing what FEEDER considers to be the true status of the window. This command orders FEEDER to alter the contents of that storage area.

CALL FEEDER (6,NPANES,NFROM,NTO)

NPANES	The number of consecutive panes to be revised.
NFROM	The number of the DECISION window pane containing the revised information for the first pane to be edited. If NPANES > 1, the remaining revised information is taken from consecutively higher numbered panes.
NTO	The number of the DECISION window pane (in storage) in which FEEDER is to place the revised information. In other words, the value of NTO is the number of the first pane to be edited. If NPANES > 1, the remaining revised information is placed in consecutively higher numbered panes.

FEEDER executes this command by performing the following sequence of tasks:

1. Bytes 1-8 of the INTERRUPTION vector are zeroed.
2. NPANES, NFROM, and NTO are inspected for reasonable values. If impossible values exist, FEEDER sets QUIT = .TRUE. and takes no further action.

3. New values for all parameters except HPMFN(1,N) are moved to each of the panes being edited. The exception of the HPMFN(1,N) parameter allows updating of corresponding main frame data to be performed either before or after the updating of DECISION data.

4. The QPANY(1,N) and QPTIMG(1,N) flags are reset appropriately for every pane in the DECISION window.

5. Flag summaries and pointers in the INTERRUPTION vector are reset to reflect the contents of the revised DECISION window.

6. The DECISION window is displayed in its edited form.

(Programming note: **Any** pane of the DECISION window may be used by a calling program to build an edited version of a pane. The proper contents of the building area will be displayed after FEEDER has completed the update. But be sure to first save any useful values that will be wiped out.)

#### F. The "update-main-frame-data" command

This command allows a calling program to edit main frame data corresponding to one or more panes of the DECISION window. Main frame data, which is supplied only by request, is kept by FEEDER in its own storage area. This command orders FEEDER to alter the contents of that area.

CALL FEEDER (7,NVECTS,NVECT1,NPANEL)

NVECTS	The number of consecutive data vectors (and corresponding flag vectors) which contain the edited version of the main frame data.
NVECT1	The number of the first data vector (as well as of the first flag vector) which contains the edited main frame data. If NVECTS > 1, the remaining edited data is taken by FEEDER from consecutively higher numbered vectors.
NPANEL	The number of the first DECISION window pane for which corresponding main frame data is to be modified. If NVECTS > 1, the remaining edited data replaces main frame data corresponding to consecutively higher numbered DECISION window panes.

FEEDER responds by performing the following sequence of tasks:

1. Bytes 1-8 of the INTERRUPTION vector are zeroed.
2. NVECTS, NVECT1, and NPANEL are examined for reasonable values. If impossible values are detected or if the option to omit main frame data processing had been specified, FEEDER sets QUIT = .TRUE. and takes no further action.
3. The edited main frame data is moved as specified.
4. For each pane whose corresponding main frame data has been revised, the revised main frame flags are counted and the count is placed in the HPMFN(1,N) parameter.

G. The "delete-half-block" command

This command allows a calling program to shorten a section of the DECISION window so that the number of panes agrees with the number of major frames represented. Squeezing out a half-block containing only fill data is expected to be one of the applications for this command.

CALL FEEDER (8,NPANES,NPANEL)

NPANES      The number of consecutive panes which are to be deleted from the DECISION window. NPANES is also the number of DECISION window panes for which corresponding main frame data will be deleted.

NPANEL      The number of the first DECISION window pane to be deleted.

The DECISION window displayed by FEEDER after executing this command will show:

1. The first (NPANEL-1) panes are unchanged.
2. The data formerly in pane (NPANEL + NPANES) will be in pane NPANEL, the data formerly in pane 30 will be in pane (30-NPANES), and intermediate data will also be shifted by NPANES panes.
3. NPANES new half-blocks, Decomm tapes permitting, will be at the upper end of the DECISION window.

Main frame data that FEEDER can supply will also be updated to correspond to the half-blocks in DECISION window panes 15-30.

If the values of NPANES and NPANEL make the command impossible, FEEDER will set QUIT = .TRUE. and take no other action.

#### H. The "skip-N-files" command

This command allows a calling program to bypass substantial data on a Decomm tape without causing FEEDER the labor of processing the unwanted intervening information. The command can be modified to skip the remainder of a Decomm tape, if desired.

CALL FEEDER (9,N)

Where the value of N indicates the number of files to be skipped. The response made by FEEDER varies with the value of N:

$N \leq 0$ . The current tape is dismounted and reading begins on the next tape. When control is returned, FEEDER has read enough of the next tape so that the first 16 half-blocks are in panes 15-30 of the DECISION window while the first 14 panes will be empty. No file information will be available for the HEADER window prior to the first file of the next tape, and neither will any orbit data be saved from previous files.

$N > 1$ . Decomm files are counted starting with file 0 being the file from which DECISION window pane 30 was taken prior to the file-skip command. If file N lies beyond the first file of the next tape, then FEEDER limits its response to be equivalent to a tape skip described above for  $N = 0$ . Otherwise, FEEDER proceeds as follows:

1. Files are skipped as necessary until file (N-1) is reached.
2. File information is taken (for the HEADER window) and orbit data is saved from file (N-1).
3. FEEDER proceeds more cautiously to retain the last 14 half-blocks of file (N-1) for the DECISION window. When control is returned, the first half-block from file N will be in pane 15 of the DECISION window. If options require it, orbit data will be available for file N, as well as some orbit data for the last part of file (N-1). File information from either file (N-1) or file N can be placed in the HEADER window.

## I. The "fetch-N-vectors-of-main-frame-data" command

CALL FEEDER (10,N,NPANEL,NVECT1)

- N           The number of consecutive DECISION window panes for which the corresponding main frame data is to be placed in the SCAN data window.
- NPANEL      The number of the first DECISION window pane for which corresponding data is to be moved to the SCAN data window. If  $N > 1$ , remaining data is taken corresponding to the next  $(N-1)$  higher numbered panes.
- NVECT1      The number of the data vector and of the corresponding flag vector to which values for pane NPANEL are moved. If  $N > 1$ , the remaining data and flags are moved to the next  $(N-1)$  higher numbered vectors.

If the values of N, NPANEL, NVECT1, make the command impossible, FEEDER will set QUIT = .TRUE. and take no further action.



## Subroutine ADRESS

Program Language : IBM-360 0/S Assembler Language

Date Written : June 29, 1971

Programmer : A. Buck

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Subprograms Called : None

Description : Subroutine ADRESS initializes storage  
areas supplied by subroutine GETCOR.  
Storage areas are partitioned appropriately  
and addresses of the various partitions  
are stored as required by subroutine  
FEEDER and its subservient routines.  
Partition-in-use flags, where present,  
are set to indicate the partition is not  
currently in use.

## Subroutine DECISION

Program Language : IBM-360 O/S Assembler Language  
Date Written : September 3, 1971  
Programmer : A. Buck  
Installation : University of New Hampshire  
Space Science Center  
Computer : IBM-360-H50  
Subprograms Called : None  
Description : This subroutine performs all processing of main frame data, subcom data, and flag fields contained in one data block of a Decomm tape. For each of the two major frames, subroutine DECISION:

A. Completes main frame storage partitions in accordance with the main frame data processing option in effect. In the event the short partition option is selected, the four experiment words are placed at the beginning of the allotted storage area.

B. Nearly completes the DECISION window pane. The only DECISION pane parameters not developed by the subroutine are:

QPANY Which depends on QPFILS and QPTAPS.

QPFILS Which only FEEDER can set properly.

QPTAPS Which only FEEDER can set properly.

UPTIME Which FEEDER must develop to determine the proper settings for preceding sync flags.

## Subroutine GETCOR

Program Language : Fortran IV, G-Level  
Date Written : June 29, 1971  
Programmer : A. Buck  
Installation : University of New Hampshire  
Space Science Center  
Computer : IBM-360-H50  
Dummy Arguments : None  
Description : Subroutine GETCOR contains storage areas  
to be used by subprogram FEEDER and its  
subserving routines. GETCOR is called  
during initialization and in turn calls  
subroutine ADDRESS to relay area addresses  
to FEEDER.

## Subroutine HEADEREC

Program Language : IBM-360 O/S Assembler Language

Date Written : June 4, 1971

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Subprogram Called : None

Description : Subroutine HEADEREC is designed to assist subprogram FEEDER by reformatting information in the file header record and storing the reformatted information to replace the older information in the two HEADER windows. (The term HEADER window here means an area of core storage containing information about a Decomm file in a form that is usable by a Fortran program. Most of the information is taken from the file header record.) In addition, subroutine HEADEREC computes time constants useful to other routines in analyzing orbit data records and experiment data to follow within the file.

## Subroutine ORBSTOR

Program Language : IBM-360 O/S Assembler Language

Date Written : June 23, 1971

Programmer : A. Buck

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Subprograms Called : TAPREAD

Description : Subroutine ORBSTOR assists subprogram FEEDER by processing orbit data records in accordance with user options. ORBSTOR is to be called after the header record has been processed but before further requests to TAPREAD are made. ORBSTOR calls TAPREAD to supply all orbit data records and processes these records. ORBSTOR returns control to FEEDER as soon as a record is read normally but which does not identify as an orbit data record. Then, when control is returned to FEEDER, the first data record in the file is presumably available in the buffer whose address is stored at 'ABUFFNEW' (see DSECT defined by the symbol 'CONTORG').

A. All orbit data items are scanned for sunlight-entrance times and for sunlight-exit times. These times are stored in the newer of the two HEADER windows.

B. If the user has elected the orbit data option, orbit data is stored as described in subroutine ORBWNDW.

C. If the user has elected the all item option as well as the orbit data option, all data items are stored if possible. If the user refuses the all item option, only the earliest data item and subsequent regular data items are stored.

## Subroutine ORBWNDW

Program Language : IBM-360 O/S Assembler Language  
Date Written : May 3, 1971  
Programmer : A. Buck  
Installation : University of New Hampshire  
Space Science Center  
Computer : IBM-360-H50  
Subprograms Called : None  
Description : This subroutine assists subroutine FEEDER by handling all requests for OSO-H orbit data (Ephemeris) values. Subroutine ORBWNDW locates, where possible, and returns a set of 4 values for each of 22 orbit data parameters. These 4 values correspond to a set of times which are centered as nearly as practical about a requested time. (All times are seconds of 20th century in double precision, floating point format.)

Data is returned by ORBWNDW in a contiguous storage area called the 'ORBIT data window'. The window area is 340 bytes in length and must originate in a double word boundary.

OSO-H orbit data (i.e., Ephemeris data) applying to a particular experimenter tape file is located on tape records at the beginning of the file and preceding records containing experiment data. When the orbit data records are read, the FEEDER subprogram converts the information to IBM-360 format and stores it for subsequent recall. (If the user fails to specify the orbit data options, the FEEDER subprogram will merely scan the orbit data for sunlight-entrance and exit times. No storage occurs and no recall is possible.) Since much of the OSO-H 3 data is read out at a time after

it was recorded, the FEEDER subprogram must retain some orbit data from the previous file to ensure providing orbit data corresponding to the time data was recorded.

Orbit data is recalled by specifying a time  $T$  near which values for orbit parameters are requested. Let  $[T(1), T(2), \dots, T(N)]$  be the times for which orbit data is stored. Subprogram FEEDER will, if possible, return 4 orbit data values for all orbit data values for all parameters for the set of times  $[T(I), T(I+1), T(I+2), T(I+3)]$  which most nearly brackets time  $T$ . In particular:

A. If the user has failed to specify the orbit data processing option, FEEDER will:

1. Print a message indicating the conflict.
2. Set a flag indicating no orbit data returned.
3. Set a flag indicating program should be aborted.

B. If  $T$  is less than  $T(1)$ , or if  $T$  is greater than  $T(N) + 59999$  msec., FEEDER will set a flag indicating no data returned.

C. If  $T$  is less than  $T(2)$  but not less than  $T(1)$  FEEDER will:

1. Return values corresponding to  $T(1), T(2), T(3), T(4)$ .
2. Set a flag indicating times are not centered.

D. If  $T$  is greater than  $T(N-1)$  but less than  $[T(N) + 60000$  msec], FEEDER will:

1. Return values corresponding to  $T(N-3), T(N-2), T(N-1), T(N)$ .
2. Set a flag indicating times are not centered.

E. If  $T$  is not less than  $T(2)$  and not greater than  $T(N-1)$ , FEEDER will return sets of values corresponding to

Set of times  $(T(I), T(I+1), T(I+2), T(I+3))$  where  $T$  exceeds  $T(I+1)$  but does not exceed  $T(I+2)$ .

In all, 22 orbit data parameters are managed by subprogram FEEDER.



## Subroutine TAPREAD

Program Language : IBM-360 O/S Assembler Language  
Date Written : August 17, 1971  
Programmer : A. Buck  
Installation : University of New Hampshire  
Space Science Center  
Computer : IBM-360-H50  
Subprograms Called : None  
Description : This subroutine assists subprogram  
FEEDER by reading and positioning OSO-H  
Experimenter Decomm tapes. The 7-track  
Decomm tapes are to be read using the  
data converter feature.

End-of-file conditions are indicated to subprogram FEEDER by means of a return code in general register 15. Return codes of 0, 4, and 8 indicate no tape marks encountered during activity; record returned is first record after last tape mark detected during activity, and end of volume (2 consecutive tape marks) detected with no record returned.

TAPREAD performs any of 4 tasks depending on the command code passed via general register 1.

REG1=0 The next block is read from the Decomm tape. If the first read encounters a tape mark, the file count is incremented, block count reset to zero, and a second read issued and checked.

REG1=4 Any Decomm tape currently open is closed and dismounted. The DDNAME parameter for the next tape is placed in the DCB and the next tape opened. The first block is read and returned.

REG1=8 The full word defined by the symbol FILESKIP (in FEEDER's control area) contains the number of files TAPREAD is to skip, beginning with the current file. The specified number of files are skipped and the first block is read from the following file and returned. If EOF limits the file skipping, the full word denoted by the symbol FILESKIP contains the number of files not skipped.

REG1=12 If possible, the 7th from the last block of the current file is read and returned.

### FEEDER Messages

Subprogram FEEDER communicates to the user through a series of messages indicating

- A. The condition of the input tapes and the data files on those tapes.
- B. Requests made by the user that cannot be met.
- C. Input/output errors.
- D. Difficulties during program execution due to anomalies in the data. (Flags are usually set and printed to help determine the problem area.)

The following is a list of the message numbers and corresponding messages printed out by FEEDER subroutines. Each message is followed by a brief explanation.

M010. HEADER RECORD FAULTY. SATELLITE ID = \_\_\_\_\_,  
YEARS = \_\_\_\_\_, EXPERIMENT NO. = \_\_\_\_\_, BEGIN  
TIME = \_\_\_\_\_, END TIME = \_\_\_\_\_

Explanation: Either the satellite identifier, the range of years, the experiment number, the interval between time and end time are inconsistent with the UNH-OSO experiment. These are considered intolerable conditions. A quit-requested flag is posted in the interruption vector.

M020. REQUEST FOR ORBIT DATA CONFLICTS WITH ORBIT OPTION.

Explanation: The user failed to elect the orbit data processing option and then later asked for orbit data. This message is printed at most once per file.

M030. REQUEST FOR ORBIT DATA WHEN NONE IN STORAGE.

Self-explanatory. This could occur if quick-look tapes were being used as input while the option to process orbit data was in effect. This message is printed once per file at most.

M040. REQUESTED ORBIT DATA NOT IN STORAGE.

Self-explanatory. The request time of data did not fall within the range of orbit data times of the orbit data currently in storage. This message is never printed more than once per file.

M050. END-OF-FILE DETECTED BY ORBIT DATA STORAGE ROUTINE.

Explanation: The organization of the experimenter tape is incorrect.

M060. I/O ERROR AND ORBIT-DATA-ID ERROR CAUSED REFUSAL OF EXPECTED ORBIT RECORD.

Explanation: The record was bypassed because of the existence of both of the named conditions.

M070. ORBIT DATA IMPROPERLY TERMINATED ON TAPE.

Self-explanatory. The orbit data may or may not have been properly filed.

M080. NO ORBIT DATA RECORDS IN FILE.

Self-explanatory.

M090. UNEXPECTED ORBIT DATA RECORD.

Explanation: The record identifier indicates the record contains orbit data. This contradicts the prediction based on the preceding record. Both records are accepted.

M100. INVALID DAY-OF-YEAR CAUSED ORBIT DATA RECORD REFUSAL.

Explanation: Day-of-year count of first data item disagrees with both possible day of year counts computed from the header record. The orbit data record is bypassed.

M110. INVALID DAY-OF-YEAR CAUSED DATA-ITEM REFUSAL. ITEM WAS SUNLIGHT ENTRANCE OR EXIT ITEM.

Explanation: An orbit data item was bypassed for the reason given. However, other orbit data items from the same record may have been successfully processed.

M120. OVERFLOW ON SUNLIGHT TIME ARRAY.

Explanation: The list of day-night exceeds planned storage. Only first four items are saved.

M130. TAPEMARK AT START OF TAPE NO. \_\_\_\_\_, SER = \_\_\_\_\_.

Explanation: Tapemark encountered at start of tape.

M150. I/O ERROR OCCURRED ON BLOCK \_\_\_\_\_ OF FILE \_\_\_\_\_ OF TAPE \_\_\_\_\_ (SER = \_\_\_\_\_) ON UNIT \_\_\_\_\_. SENSE = \_\_\_\_\_, STATUS = \_\_\_\_\_.

Explanation: Input/output error has occurred; usually a result of either bad tape or problem tape drive.

M160. NON-STANDARD BLOCK LENGTH = \_\_\_\_\_ BYTES ON BLOCK \_\_\_\_\_ OF FILE \_\_\_\_\_ OF TAPE \_\_\_\_\_ (SER = \_\_\_\_\_) ON UNIT \_\_\_\_\_.

Self-explanatory. Block of data of incorrect block size has been read.

- M170. UNEXPECTED FRAME COUNTS AT BLOCK \_\_\_\_\_.  
Self-explanatory.
- M180. TIME FIELD DAY COUNT = \_\_\_\_\_ INSTEAD OF \_\_\_\_\_  
OR \_\_\_\_\_.  
Explanation: Day of year count disagrees with both possible day-of-year counts computed from the header record.
- M190. COMMAND CODE OUT OF ALLOWABLE RANGE.  
Self-explanatory: Command code is greater than 10.
- M200. COMMAND PRECEDED INITIALIZATION COMMAND.  
Self-explanatory: Initialization command must be issued before other commands to FEEDER can be executed.
- M210. INITIALIZATION COMMAND ISSUED MORE THAN ONCE.  
SUBSEQUENT COMMAND REFUSED.  
Self-explanatory.
- M220. FEEDER TO PROCESS # \_\_\_\_\_ TAPES, (accept or reject) WASC WDS 3-10, accept or reject ATT DATA, (accept or reject) ORBIT ITEMS.  
Explanation: Message is printed after initialization command has been issued to indicate options selected by user.
- M230. INVALID DECOMM TAPE COUNT. \* 1 SUBSTITUTED.  
Explanation: Tape count may not exceed 12.
- M240. INVALID MF DATA PROCESSING OPTION CODE. 0 SUBSTITUTED.  
Explanation: Code must be 0, 1, or 2.
- M250. INVALID ORBIT OPTION CODE. 0 SUBSTITUTED.  
Explanation: Code must be 0, 1, or 2.

- M260. INVALID PARAMETERS TO UPDATE DECISION WINDOW.  
Explanation: Call arguments invalid.
- M270. INVALID PARAMETERS TO UPDATE MAIN FRAME DATA.  
Explanation: Call arguments invalid.
- M280. INVALID PARAMETERS TO DELETE HALFBLOCKS.  
Explanation: Call arguments invalid.
- M290. FILESKIP TERMINATED BY EOVS AT END OF FILE \_\_\_\_\_.  
Self-explanatory.
- M300. INVALID PARAMETERS TO FETCH MAIN FRAME DATA.  
Explanation: Call arguments invalid.
- M310. ALL PANES ARE VACANT. END-OF-DATA FLAG IS \_\_\_\_\_.  
Explanation: Indicates no data in decision window.

M220. FEEDER TO PROCESS 1 TAPES, ACCEPT ALL MF DATA, ACCEPT DSM WDS, ACCEPT WASC WDS 3-10, ACCEPT ATT DATA, ACCEPT REG ORBIT ITEMS.

\*\*\*\* M140. BLOCK 1 READ FROM FILE 1 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 2 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 3 OF TAPE 1 (SER=EXD224).

M170. UNEXPECTED FRAME COUNTS AT BLOCK 140.

\*\*\*\* M140. BLOCK 1 READ FROM FILE 4 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 5 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 6 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 7 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 8 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 9 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 10 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 11 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 12 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 13 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 14 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 15 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 16 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 17 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 18 OF TAPE 1 (SER=EXD224).

M100. INVALID DAY-OF-YEAR CAUSED ORBIT DATA RECORD REFUSAL.

\*\*\*\* M140. BLOCK 1 READ FROM FILE 19 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 20 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 21 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 22 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 23 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 24 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 25 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 26 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 27 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 28 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 29 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 30 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 31 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 32 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 33 OF TAPE 1 (SER=EXD224).

\*\*\*\* M140. BLOCK 1 READ FROM FILE 34 OF TAPE 1 (SER=EXD224).

Actual FEEDER Messages  
as they appear in computer output



M220. FEEDER TO PROCESS 8 TAPES, ACCEPT ALL MF DATA, ACCEPT DSM WDS, ACCEPT WASC WDS 3-10, ACCEPT ATT DATA, ACCEPT REG ORBIT ITEMS.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 1 OF TAPE 1 (SER=EXD314).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 2 OF TAPE 1 (SER=EXD314).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 3 OF TAPE 1 (SER=EXD314).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 4 OF TAPE 1 (SER=EXD314).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 5 OF TAPE 1 (SER=EXD314).  
 M170. UNEXPECTED FRAME COUNTS AT BLOCK 20.  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 6 OF TAPE 1 (SER=EXD314).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 7 OF TAPE 1 (SER=EXD314).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 8 OF TAPE 1 (SER=EXD314).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 9 OF TAPE 1 (SER=EXD314).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 10 OF TAPE 1 (SER=EXD314).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 11 OF TAPE 1 (SER=EXD314).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 12 OF TAPE 1 (SER=EXD314).  
 M090. UNEXPECTED ORBIT DATA RECORD.  
 M090. UNEXPECTED ORBIT DATA RECORD.  
 M090. UNEXPECTED ORBIT DATA RECORD.  
 M090. UNEXPECTED ORBIT DATA RECORD.  
 M090. UNEXPECTED ORBIT DATA RECORD.  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 13 OF TAPE 1 (SER=EXD314).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 14 OF TAPE 1 (SER=EXD314).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 15 OF TAPE 1 (SER=EXD314).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 16 OF TAPE 1 (SER=EXD314).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 1 OF TAPE 2 (SER=EXD315).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 2 OF TAPE 2 (SER=EXD315).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 3 OF TAPE 2 (SER=EXD315).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 4 OF TAPE 2 (SER=EXD315).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 5 OF TAPE 2 (SER=EXD315).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 6 OF TAPE 2 (SER=EXD315).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 7 OF TAPE 2 (SER=EXD315).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 8 OF TAPE 2 (SER=EXD315).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 9 OF TAPE 2 (SER=EXD315).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 10 OF TAPE 2 (SER=EXD315).  
 M040. REQUESTED ORBIT DATA NOT IN STORAGE.  
 \*\*\*\* M140. BLOCK 1 READ FROM FILE 11 OF TAPE 2 (SER=EXD315).

Actual FEEDER Messages  
 as they appear in computer output

## Section 2

### PASS-I

	<u>Page</u>
I. Introduction	2-1
II. Macro Systems Flowchart	2-3
III. Job Stream Schematic	2-4
IV. Job Control Language	2-5
V. Data Processing Program (OSOP103C)	2-7
VI. Subroutine CALIB (X,NSUM)	2-25
VII. Subroutine CALNDR (USOC,IYR,IMO,IDA,IHR,IMN,SEC)	2-27
VIII. Subprogram FEEDER	2-28
IX. Subroutine FORTIM (IRUNDA,IRUNT)	2-29
X. Subroutine FPUT (IDSN,ICMD,IRSLT,IOVEC)	2-30
XI. Subroutine FTWRIT (IDSN,ICMD,IEXT,IRSLT,OUTBLK)	2-32
XII. Subroutine MOVER (AREAL,IDISP1,AREA2,IDISP2,LEN)	2-35
XIII. Subroutine OUTPUT	2-37
XIV. Subroutine PLOUT	2-49

XV.	Subroutine PUZZLE	2-50
XVI.	Function QUASH (HAR,HMASK,LEN)	2-51
XVII.	Subroutine SUMARY (KFSUMY,HFSUMY,KNFDSN,KAFDSN, KJFDSN,I)	2-52
XVIII.	Subroutine SUMCAL (KFSUMY,KTSUMY)	2-53
XVIX.	Plot Production Program (OSOP104A)	2-54
XX.	Subroutine ENPLTA	2-55
XXI.	Subroutine FGET (IDSN,IRSLT,IOVEC)	2-56
XXII.	Subroutine INITPL	2-58
XXIII.	Subroutine OUTLNE	2-59
XXIV.	Description of INPUT	2-60
XXV.	Description of OUTPUT	2-71
XXVI.	Sample OUTPUT	2-88

## I. Introduction

There are six possible options to this PASS. The options can be executed in any combination depending on the parameters specified on the control card. The options are as follows:

- A. Engineering plot.
- B. Line summary.
- C. Calibration cards.
- D. X-ray tape.
- E. Good time cards.
- F. Main frame data tape.

The engineering plots are constructed for the various log rate meters which monitor the counting rates of the scintillators comprising the instrument. Temperature, voltages, and status are also monitored. The output of the program is stored on a Memorex-3660 disk and then 7-track plot tapes are created in a second job step. The plots contain status information and housekeeping information rates for the central detector, cup, and slab.

Each line in the one-line summary represents information for three minutes of satellite data. This corresponds to the accumulation of a complete gamma ray spectrum. Time, rates, status, sync errors, and warnings are printed.

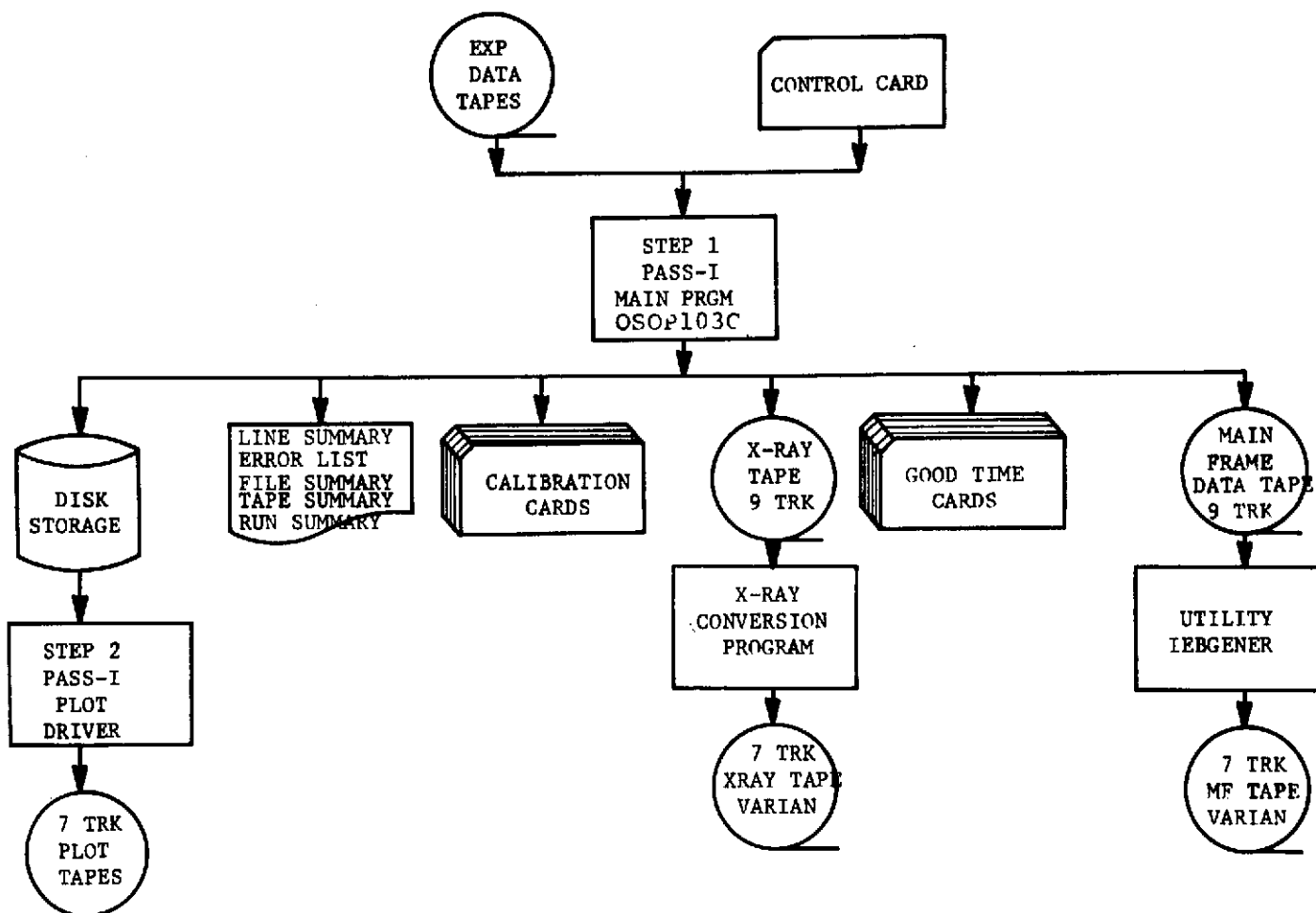
The calibration data consists of the central channels of the three gamma peaks in the calibration spectrum interpolated from a quadratic least squares fit to each peak. The total counts in each peak and the midpoint of the time interval during which the calibration was accumulated are also included.

The X-ray data is obtained from the Experimenter tapes with detailed spin and timing information to allow exact time correlations of the X-ray data with solar activity indices from other experiments. This data is stored on a 9-track tape which is Fortran compatible and may be used on the IBM-360-H50. A 7-track reformatted version is also generated for use on a Varian-620/i minicomputer.

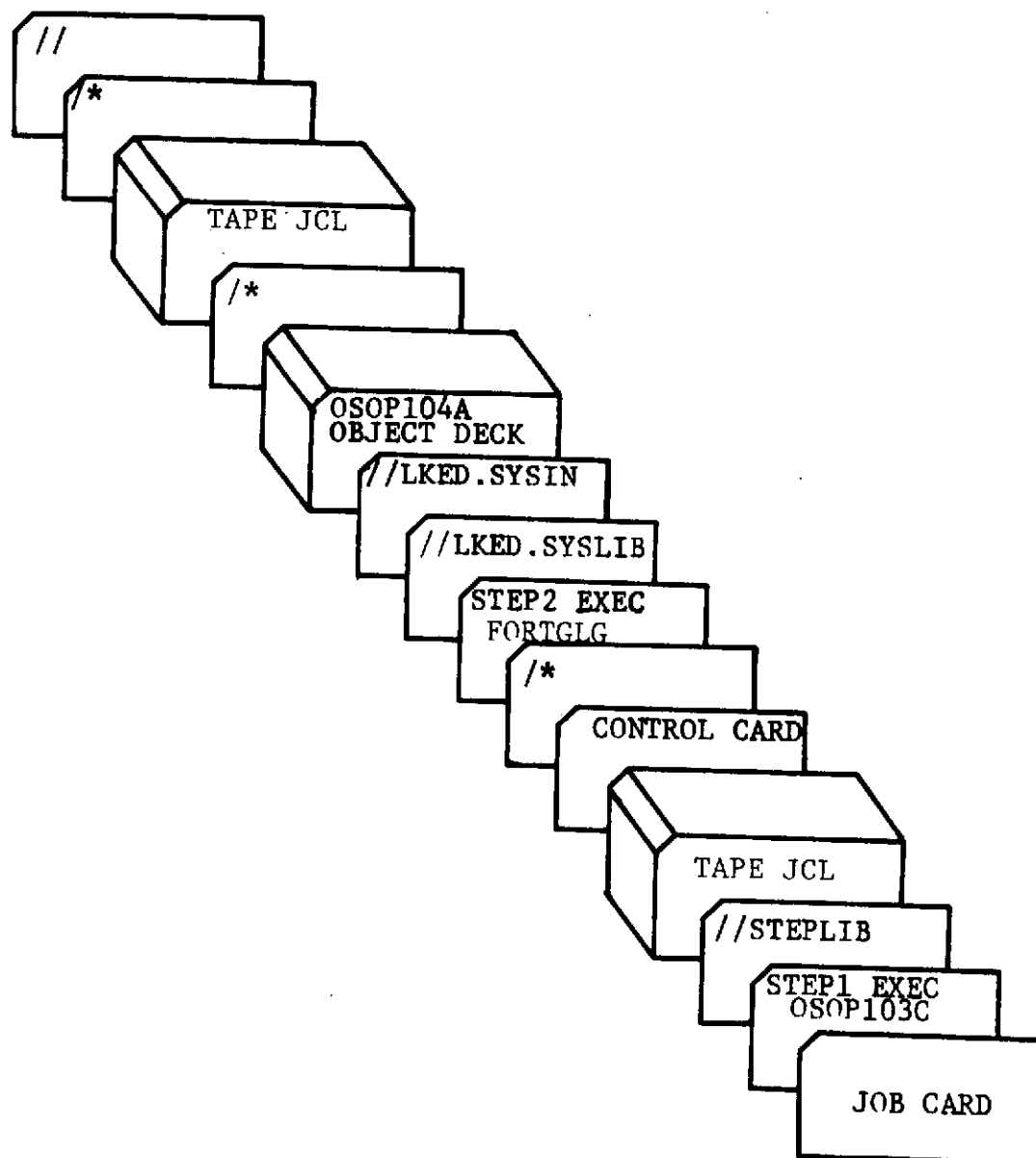
The good time cards consist of begin and end midpoint of the accumulation times for good continuous data. Only data collected when the central detector was on and with no time gaps is considered "good". These cards are later used as input into Pass-III.

Main frame data for each quadrant scan collected when power was on is written onto a 9-track tape along with analog data and read-out time. The tape is then converted to 7-track for use on the Varian-620/i minicomputer with Tektronix-4010 display unit interface.

## II. Macro Systems Flowchart (PASS-I)



## III. Job Stream Schematic (PASS-I)



## IV. JOB CONTROL LANGUAGE - PASS-I

```

// T548,,20,,,,,60),'CROTEAU,PASS-I',CLASS=E,TIME=10,REGION=240K,
// MSGLEVEL=(1,1)
//STEP1 EXEC PGM=DSOP103C,TIME=10
//STEPLIB DD DSN=USER.T548.SUBLIB,DISP=SHR
//DECOM1 DD UNIT=TAPE7,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD224
//DECOM2 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD307
//DECOM3 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD308
//DECOM4 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD346
//DECOM5 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD347
//DECOM6 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD329
//DECOM7 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD330
//DECOM8 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD317
//DECOM9 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD318
//DECOM10 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD319
//DECOM11 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD029
//DECOM12 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD030
//FPUT01 DD UNIT=2400,DISP=(NEW,KEEP),LABEL=(1,SL),
//          DSNAME=XRAY,DCB=(RECFM=FB,LRECL=76,BLKSIZE=1520),
//          VOLUME=SER=CC0920
//FPUT02 DD UNIT=2400,DISP=(NEW,KEEP),LABEL=(1,SL),
//          DSN=MFDATA,DCB=(RECFM=FB,LRECL=908,BLKSIZE=908),
//          VOLUME=SER=MFD020
//FTWT01 DD UNIT=SYSDA,DSN=88PLOTOM,DISP=(NEW,PASS),
//          DCB=(BLKSIZE=6500),SPACE=(TRK,(200,100),RLSE)
//FT19F001 DD UNIT=SYSDA,DISP=(NEW,PASS),DSN=88NPLOT,
//          DCB=(RECFM=FB,LRECL=24,BLKSIZE=24),SPACE=(TRK,1)
//MSGDD DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//FT06F001 DD SYSOUT=A,DCB=(RECFM=UA,BLKSIZE=133)
//FT07F001 DD SYSOUT=B,DCB=(RECFM=F,BLKSIZE=80)
//FT15F001 DD SYSOUT=A,DCB=(RECFM=UA,BLKSIZE=133)

```



```
//FT16F001 DD SYSOUT=A,DCB=(RECFM=UA,BLKSIZE=133)
//FT17F001 DD SYSOUT=A,DCB=(RECFM=UA,BLKSIZE=133)
//FT18F001 DD SYSOUT=A,DCB=(RECFM=UA,BLKSIZE=133)
//FT20F001 DD SYSOUT=A,DCB=(RECFM=UA,BLKSIZE=133)
//FT05F001 DD *
  1 1 1 1 1 0 0 1 1 1 1 0 199
/*
//STEP2 EXEC PLOTGLG,REGION=126K,TIME.GO=10
//LKED.SYSLIB DD DSN=USER.T317.SUBLIB,DISP=SHR
//          DD DSN=USER.T548.SUBLIB,DISP=SHR
//          DD DSN=SYS1.FORTLIB,DISP=SHR
//          DD DSN=USER.PLOTLIB,DISP=SHR
//          DD DSN=USER.SSPLIB,DISP=SHR
//LKED.SYSLMOD DD DSN=&GOSET(SUE2)
//LKED.SYSIN DD *
```

\*\*\* OSOP104A OBJECT DECK \*\*\*

```
/*
//GO.FT32F001 DD UNIT=(TAPE7,,DEFER),DSN=PLOTTER,DISP=(NEW,KEEP),
//          LABEL=(1,BLP),
//          DCB=(DEN=1,TRTCH=ET,BUFNO=1,RECFM=U,BLKSIZE=1024),
//          VOL=SER=PLT001
//GO.FT32F002 DD UNIT=AFF=FT32F001,DSN=PLOTTER,DISP=(NEW,KEEP),
//          LABEL=(1,BLP),
//          DCB=(DEN=1,TRTCH=ET,BUFNO=1,RECFM=U,BLKSIZE=1024),
//          VOL=SER=PLT002
//GO.FT32F003 DD UNIT=AFF=FT32F001,DSN=PLOTTER,DISP=(NEW,KEEP),
//          LABEL=(1,BLP),
//          DCB=(DEN=1,TRTCH=ET,BUFNO=1,RECFM=U,BLKSIZE=1024),
//          VOL=SER=PLT003
//GO.FT32F004 DD UNIT=AFF=FT32F001,DSN=PLOTTER,DISP=(NEW,KEEP),
//          LABEL=(1,BLP),
//          DCB=(DEN=1,TRTCH=ET,BUFNO=1,RECFM=U,BLKSIZE=1024),
//          VOL=SER=PLT004
//GO.FT32F005 DD UNIT=AFF=FT32F001,DSN=PLOTTER,DISP=(NEW,KEEP),
//          LABEL=(1,BLP),
//          DCB=(DEN=1,TRTCH=ET,BUFNO=1,RECFM=U,BLKSIZE=1024),
//          VOL=SER=PLT005
//GO.FT32F006 DD UNIT=AFF=FT32F001,DSN=PLOTTER,DISP=(NEW,KEEP),
//          LABEL=(1,BLP),
//          DCB=(DEN=1,TRTCH=ET,BUFNO=1,RECFM=U,BLKSIZE=1024),
//          VOL=SER=PLT006
//GO.FT32F007 DD UNIT=AFF=FT32F001,DSN=PLOTTER,DISP=(NEW,KEEP),
//          LABEL=(1,BLP),
//          DCB=(DEN=1,TRTCH=ET,BUFNO=1,RECFM=U,BLKSIZE=1024),
//          VOL=SER=PLT007
//GO.FT19F001 DD DSN=&&NPLOT,DISP=(OLD,DELETE),DCB=(RECFM=F,
//          LRECL=24,BLKSIZE=24)
//GO.FGET01 DD DSN=&&PLOTCOM,DISP=(OLD,DELETE),DCB=(RECFM=F,
//          LRECL=6500,BLKSIZE=6500)
//GO.SYSUDUMP DD SYSOUT=A
//GO.SYSIN DD *
/*
//
```

## V. Data Processing Program (OSOP103C)

Program Language : Fortran IV, H-Level  
Date Written : January 1972  
Programmer : A. Buck  
Installation : University of New Hampshire  
Space Science Center  
Computer : IBM-360-H50  
Subprograms Called : FORTIM, FEEDER, MOVER, PLOUT, SUMARY,  
SUMCAL, FTWRIT, FPUT, CALNDR, QUASH,  
PUZZLE, OUTPUT

The purpose of this program is to control the flow of data, to select and edit valid sections of data, and to output these selected sections according to desired options.

The basic functions performed by the program are as follows:

- A. The initialization (and reinitialization) of variables referenced by OUTPUT.
- B. The input of experiment data using FEEDER.
- C. The rejection of data due to I/O errors, illogical data, indeterminate data, and high voltage off.
- D. The assignment of time sequence identifiers to each major frame. Locating and identifying the start of the background scan and start of solar scan.
- E. The selection of sets of consecutive input major frames for processing (optimally, 12 major frames comprising 1 full scan are selected)
  - 1. Same file
  - 2. Same scan mode
  - 3. Same calibration status

4. Not over three minutes of data.
  5. Followed by start of background scan.
- F. The production of data anomaly messages.
- G. The production of summaries for each file, for each tape, and for the entire run.
- H. The output of:
1. Data anomaly messages.
  2. File summary.
  3. Plot frame count.
  4. Return code.
  5. Engineering data tables to be plotted.
- I. The call to OUTPUT.

The inputs to this program are:

- A. The experimenter data tape.
- B. The option specification card (otherwise known as the control card).

The outputs of this program are:

- A. For each run
1. FEEDER messages - printer.
  2. Data condition messages - printer.
  3. File summaries - printer.
  4. Count of plot frames - disk.
  5. Return code.

## B. Optional Outputs

1. Engineering plot tables - disk.
2. Line summaries - printer.
3. Calibration cards - card punch.
4. X-ray data - tape.
5. Good time cards - card punch.
6. Main frame data - tape.

Subprogram FEEDER uses common storage to provide this program with the input data. This program, in turn, provides subse-  
vient routines with the information they need to examine the  
selected data through common storage. In the following pages,  
the variables in common storage are described to help the reader  
follow program logic.

## A. Labelled common areas

1. COMMON/HEADER/ : description of variables can  
be found in Section 1 of this report.
2. COMMON/DECIDE/ : description of variables can  
be found in Section 1 of this report.
3. COMMON/INTRUP/ : description of variables can  
be found in Section 1 of this report.
4. Extension to COMMON/INTRUP/ : the following is  
a list and description of the variables that form  
the extension to the interruption vector.

### a. X-ray tape parameters

(Note: all variables beginning with an H are integer half words;  
variables beginning with a U are double precision floating  
point words; variables beginning with a Q are logical bytes  
(8 bits); all other variables follow the explicit Fortran con-  
ventions.)

HYRR1	: year of 1st major frame.
HYRR2	: year of 2nd major frame.
HDAY1	: day of 1st major frame.
HDAY2	: day of 2nd major frame.
HHR1	: hour of 1st major frame.

HHR2 : hour of 2nd major frame.  
HMIN1 : minutes of 1st major frame.  
HMIN2 : minutes of 2nd major frame.  
SEC1 : seconds of 1st major frame.  
SEC2 : seconds of 2nd major frame.  
HSTAT1 : status of 1st major frame.  
HLSOL : solar 7.5 - 15 keV (interpreted counts).  
HML1SOL : solar 15 - 30 keV (interpreted counts).  
HM2SOL : solar 30 - 60 keV (interpreted counts).  
HHSOL : solar 60 - 120 keV (interpreted counts).  
HLBKD : background 7.5 - 15 keV (interpreted counts).  
HMLMKD : background 15 - 30 keV (interpreted counts).  
HM2BKD : background 30 - 60 keV (interpreted counts).  
HHBKD : background 60 - 120 keV (interpreted counts).  
HSQT1 : solar quadrant time of 1st major frame.  
HSQT2 : solar quadrant time of 2nd major frame.  
HBQT1 : background quadrant time of 1st major frame.  
HBQT2 : background quadrant time of 2nd major frame.  
HSTAT2 : status word of 2nd major frame.  
ASPIN1 : spin rate of 1st major frame.  
ASPIN2 : spin rate of 2nd major frame.  
AZZ1 : azimuth encoder time of 1st major frame.  
AZZ2 : azimuth encoder time of 2nd major frame.  
HELCAL : electronic calibration: 1 = on 2 = off.  
HRAD : radiation calibration: 1 = on 2 = off.  
HDANT : day - night indicator: 1 = day 2 = night.  
HFIL : extra half-word.

## b. Main frame data tape parameters:

HYR : year of read-out.  
 HMO : month of read-out.  
 HDA : day of read-out.  
 HJUL : Julian day of read-out.  
 HHR : hour of read-out.  
 HMN : minutes of read-out.  
 HSE : seconds of read-out.  
 HAN(60) : analog subcom words (10 words per major frame).  
 HMF DAT(384) : main frame data (raw counts).  
 HTGAP : time gap indicator: '1' = time gap '0' = no time gap.  
 HFCNT : # major frames processed in section.  
 HEXTRA : extra word.

c. Other arrays and variables provided in  
COMMON/INTRUP/

ASWSOD	IMAGGY	JXPANE(15)	LPFPX	QSOL1
BG(4)	IMFD	KACDSN	LSDOY	QSOL2
FRCTN	IMG1	KADDSN	LSTPLT	QSSCHG
HLCAL(16)	IMINH	KAMDSN	LXAPLT(16)	QSWCON
HLMODE(16)	IOPT	KAUSEX	LXPANE	SWSOD
HLSOLR(16)	IPLT	KFWHSC	LXPLT(16)	THFSOD
HLSTAT(16)	IQD1	KGAPSC	MAXEND	TPFSOD
HPREV	IRCNT	KJCDSN	MPUZSC	UBGDT
HTFN(6)	IRUNDA	KJDDSN	MSEVER	UEGDT
IBKSL1	IRUNT	KJMDSN	MSGID(2)	UHF SOC
IBKSL2	ISCN1	KLOSSC	MSGINC	UPLTGO
ICALIB	ISCSTR	KLSDSN	MSGSET	ULSTAT
ICNTRO	ISHD1	KNCD SN	MPANE	
ICNTRL	ISHLDO	KNDDSN	NAQUAD	
ICOFF	ISTHLD	KNMDSN	NABOS	
IDAH	ISTPHD	KOMAND(7)	NBOS	
IDAYHD	ISTPNN	LABOS	NOSEQD	
IDNITE	ISTPL	LADOY	NOWHV	
IDN1	IXPANE	LAMODE(16)	NPFIL	
IERROR	IXRAY	LASHV	QDSCHG(15)	
IFILST	IXSEQD(3,7)	LBOS	QGAP	
IGDT	IZERO(133)	LHDECN	QGIVUP	
IGNHE	IZZ(16)	LHFBLK	QSET	
IGN1	JPXMOD(47)	LHFDOY	QSHVOF	
IHRH	JPXMTH(47)	LHFPX	QSHVON	
ILIST	JXBGND(47)	LISEND	QSLVOF	
ILSUMY	JXDIEF	LPFDOY	QSLVON	

ASWSOD : time (seconds of day) of sync word values saved.

BG(4) : array of 4 half-words containing the alphameric characters SO, BO, BK, BO.

FRCTN : contains the quotient of  $1/390.625$ . This value is used in the calculation of clock time.

HLCAL(16) : calibration status code. HDCAL(1)  $\equiv$  HLCAL(2):

- 0  $\rightarrow$  no calibration.
- 1  $\rightarrow$  radiation calibration in progress.
- 2  $\rightarrow$  electronic calibration in progress.
- 4  $\rightarrow$  no calibration in progress but immediately follows a calibration.

(Note: code 4 appears once only for the half-scan following a calibration.)

HLMODE(16) : scan mode indicator. HDMODE(1)  $\equiv$  HLMODE(2):

- 6  $\rightarrow$  full scan.
- 2  $\rightarrow$  intermediate scan.
- 1  $\rightarrow$  short scan.
- 0  $\rightarrow$  do not know.

HDMODE(J) = half-scan length (major frames) of mode indicated by J'th beginning of scan in panes 15 - 29.

HLMODE(1) is used to save most recent value of HDMODE from previously processed major frames.

HLSOLR(16) : solar - background indicator HDSOLR(1)  $\equiv$  HLSOLR(2).

HDSOLR(J) = 0 or 1 according as solar or background was indicated in J'th beginning of scan in panes 15 - 29.

HLSOLR(1) is used to save the most recent value of HDSOLR from the previously processed major frames.

HLSTAT(16) : Status bits other than solar - background.  
HDSTAT(1)  $\equiv$  HLSTAT(2):

HDSTAT(J) = status bits for J'th beginning of scan in panes 15 - 29.

HLSTAT(1) = used to save the most recent value of HDSTAT from previously processed major frames.

HLSTAT(1) = -1 implies that there are meaningless values is HLSOLR(1), HLMODE(1), LXPLT(1), HLCAL(1), ULSTAT, LSDOY, LSSOD.

HPREV : -1 indicates there is no previous status.

HTFN(6) : array containing file and tape delimiters.

IBKSL1 : refers to the scan mode of the beginning good time being saved in the good time card section of OUTPUT: = 1 for SO (solar) = 4 for BO (both).

IBKSL2 : refers to the scan mode of the ending good time being saved in the good time card section of OUTPUT: = 3 for BK (background) = 4 for BO (both).

ICALIB : = 1 indicates that calibration cards are desired.

ICNTRO : status line hold area for central - detector on - off indication: 1 = on 2 = off.

ICNTRL : alphameric representation of the central - detector on - off indication for status line. ON or OFF.

ICOFF : hold area for central detector on - off indication in the plot section of OUTPUT.

IDAYH : saved day of last printed status line.

IDAYHD : day - night hold area in x-ray section of OUTPUT.

IDNITE : hold area for the numeric representation of day - night condition for status line: 1 = day 2 = nite.

IDN1 : alphameric representation of day - night indication for status line: DAY or NITE.

IERROR : = 1 if a problem occurred in OUTPUT that requires run termination by the main program. (e.g. I/O error occurred in tape write).

IFILG : filler.

IFILR : filler.

IFILST : = 1 to indicate a file start which signifies that a status line should be printed.



IGDT : = 1 to indicate that good time cards are desired.

IGNHL : hold area for gain indication in line summary section of OUTPUT: 1 = hi 2 = lo.

IGN1 : alphameric representation for the gain indication to be printed in the status line: HI or LO.

IHRH : saved hours of day printed in the last status line.

ILIST : = 1 to indicate line summary is desired.

ILSUMY : = 1 to force program to save day, hour, minutes and seconds of the status line.

IMAGGY : hold area for magnetometer-gyro reference to be printed in status line: 1 = magnetometer 2 = gyro.

IMFD : = 1 to indicate that a main frame data tape is desired.

IMG1 : alphameric representation of the magnetometer-gyro reference to be printed in status line. MAG or GYRO.

IMINH : saved minutes of status line.

IOPT : 0 = combination of options desired.  
 1 = engineering plot only.  
 2 = line summary only.  
 3 = calibration cards only.  
 4 = x-ray tape only.  
 5 = good time cards only.  
 6 = main frame data tape only.  
 IOPT is read in from the control card.

IPLOT : = 1 to indicate the engineering plot is desired.

IQD1 : alphameric representation of the normal - alternate quadrant to be printed in status line: NORM or ALT.

IRCNT : the number of main frame data tape records written.

IRUNDA : run date (argument used in call to FORTIM).

IRUNT : run time (argument used in call to FORTIM).  
 ISCN1 : alphameric representation of the scan mode to  
 print in the status line. F, I, or S.  
 ISCSTR : hold area for the scan mode to be printed in  
 status line: 1 = FULL 2 = INTER 3 = SHORT.  
 ISHDL : alphameric representation of the shield on - off  
 indication to be printed in the status line: ON or OFF.  
 ISHLDO : numeric representation of the shield on - off  
 indication to be saved with the printing of the  
 status line: 1 = ON 2 = OFF.  
 ISTHLD : status hold area in x-ray tape section of OUTPUT.  
 ISTPHD : step number hold area in plot section of OUTPUT.  
 ISTPNN : step number hold area in line summary section of OUTPUT.  
 ISTPl : step number print area for status line.  
 IXPANE : index of first pane of current section.  
 IXRAY : = 1 to indicate that the x-ray tape option is  
 desired.  
 IXSEQD(3,7) : for J = 1, NOSEQD  
 IXSEQD(I,J) = index of I'th pane, J'th digital  
 sequence. (An index of 0 implies fill is to  
 be used.)  
 IZERO(133) : an array equivalent to various other arrays  
 for initialization purposes.  
 IZZ(16) : array equivalent to IXAPLT(15).  
 JPMOD(47) : array used to express scan mode identification  
 code as a function of monitor array index:  
 0 → not determined.  
 1 → full.  
 2 → intermediate.  
 3 → short.  
 4 → cannot tell.

JPXMTH(47) : array used to express method of assigning scan mode identification as a function of monitor array index.

A method code of:

- 0 implies sheer guessing due to complete lack of scan mode clues.
- 1 implies mode determined from the first of two continuous status words.
- 2 implies mode determined throughout a complete scan from valid status word within the scan.
- 3 implies mode determined filling an identification gap by using the HGAPOS and HIDFIL arrays.
- 4 implies mode determined from analog data.
- 5 implies mode determined by free wheeling.

JXBGND(47) : array used to express start of complete scan as a function of monitor array index:

JXBGND(J) = 1 if a complete scan starts in major frame corresponding to monitor index = J  
= 0 otherwise

JXDIFF : index conversion factor = (JPXID(N)-N).

JXPANE(15) : pane indices of frames containing status words.

KACDSN : first calibration print data set (not used after program maintenance).

KADDSN : first dump data set.

KAMDSN : first message print data set.

KAUSEX : dump necessity indicator:

0 → do not dump.

1,3 → status word loss.

2,3 → analog sync loss.

KFWHSC : message severity code of '1'.

KGAPSC : message severity code of '2'.

KJCDSN : increment for calibration print data set.  
 KJDDSN : increment for dump data set.  
 KJMDSN : increment for message print data set.  
 KLOSSC : message severity code '2'.  
 KLSDSN : line summary data set (logical unit 6).  
 KNCDSN : last calibration print data set.  
 KNDDSN : last dump data set.  
 KNMDSN : last message print data set.  
 KOMAND(7) : area that contains the data requirements desired by the user and requested by FEEDER.  
 LABOS : count of analog sync words recognized in section.  
 LADOY : time (day of year) of sync word values saved.  
 LAMODE(16) : scan mode indicator: JAMODE(1)  $\equiv$  LAMODE(2)  
                   JAMODE(J) contains the mode (1 $\rightarrow$ full, 2 $\rightarrow$ intermediate, 3 $\rightarrow$ short, 4 $\rightarrow$ not available) indicated in ASSC 1-2 following the analog sync word whose plot index is given in JXAPLT(J).  
                   LAMODE(1) is used to save the mode corresponding to LXAPLT(1).  
                   Frame containing scan mode must immediately (in time) follow sync word frame.  
 LASHV : high voltage indicator for previous section (0 = off, 1 = on).  
 LBOS : count of status words in current section.  
 LHDECN : Decomm run number \* 100 + file count.  
 LHFBLK : block number of last major frame handled (whether or not included in line summaries).  
 LHFDOY : time (day of year) of last frame handled (whether or not included in line summaries).  
 LHFPX : plot index of last major frame handled (whether or not included in line summaries).  
 LISEND : maximum plot index for which entries allowed.

LPFDOY : time (day of year) of the last major frame processed (included in line summaries).  
 LPFPX : plot index of last major frame processed (included in line summaries).  
 LSDOY : time (day of year) of saved status parameter values.  
 LSTPLT : plot index of the last pane to have been processed.  
 LXAPLT(16) : plot indices: JXAPLT(1) = LXAPLT(2).  
           LXAPLT(1) = -10,000+ no plot index saved from previous data.  
 LXPANE : index of last pane of current section.  
 LXPLT(16) : plot index of frame containing status word: JXPLT(1) = LXPLT(2).  
           JXPLT(J) = plot index for J'th beginning of scan in panes 15 - 29.  
           LXPLT(1) is used to save most recent value of JXPLT from previously processed major frames.  
 MAXEND : GLB of pane indices after current section due to status word synchronization.  
 MPUZSC : message severity code of '0'.  
 MSEVER : message severity codes.  
 MSGID(2) : alphameric message set fields.  
 MSGINC : message set count increment.  
 MSGSET : message set counter.  
 MXPANE : number of last pane for current section due to status word sync requirements.  
 NAQUAD : hold area for normal - alternate quadrant to be printed in status line: 1 = normal 2 = alternate  
 NABOS : count of analog sync words recognized or count of sync words for analog word 1 in panes 15 - 29.  
 NBOS : count of status words recognized in window or count of beginning of scans in panes 15 - 29.  
 NOSEQD : count of sequences selected for x-ray processing or number of digital sequences identifiable in current section.

NOWHV : high voltage indicator for current section  
(0 = on 1 = off).

NPFIL : number of entries in IHORB.

QDSCHG(15) : QDSCHG(J) = .TRUE. implies the status indicated  
for the J'th beginning of scan in panes 15 - 29  
represents a change from the status in the pre-  
vious beginning of scan. [The solar-background  
bit and the low-order dead time bits are ignored  
in the setting of QDSCHG(J).]

QGAP : .TRUE. indicates time gap is present.

QGIVUP : program perplexity indicator.  
.TRUE. → dump, do not process.

QSET : = .TRUE. indicates good time card information to  
be punched.

QSHVOF : shield or central detector OFF condition observed  
(in DECISION window).

QSHVON : shield or central detector ON condition observed  
(in DECISION window).

QSLVOF : low voltage OFF condition observed (in DECISION  
window).

QSLVON : low voltage ON condition observed (in DECISION  
window).

QSOL1 : = .TRUE. if normal background quadrant has been  
processed for main frame data section of OUTPUT  
and solar quadrant is now to be processed.

QSOL2 : = .TRUE. if abnormal (time gap) background  
quadrant has been processed and solar quadrant  
is now to be processed. (Main frame data sec-  
tion of OUTPUT.)

QSSCHG : any-status-change-in-section indicator.

QSWCON : status word continuity indicator.

SWSOD : time (seconds of day) of saved status parameter  
values.

THFSOD : time (seconds of day) of last major frame handled  
(whether or not included in line summaries).

TPFSOD : time (seconds of day) of last major frame processed (included in line summaries).

UBGDT : beginning good time in seconds of century.

UEGDT : ending good time in seconds of century.

UHFSOC : time (seconds of century) of last major frame handled (whether or not included in line summaries).

UPLTGO : time (seconds of century) of first plot element.

ULSTAT : time (seconds of century) of saved status parameter values.

B. Unlabelled common area:

HHOUR(1300) : hour of each major frame: -2 = no value.

HDAY(1300) : day-night status: 0 = no value 1 = day 2 = night.

HNALT(1300) : normal-alternate quadrant: 0 = no value 1 = normal 2 = alternate.

HMAG(1300) : magnetometer-gyro reference: 0 = no value 1 = magnetometer 2 = gyro.

HCOFF(1300) : central detector: 0 = no value 1 = on 2 = off.

HSOFF(1300) : shield: 0 = no value 1 = on 2 = off.

HTEL(1300) : electronic temperature: -2 = no value.

HDET(1300) : detector temperature: -2 = no value.

HCNTR(1300) : central detector voltage: -2 = no value.

HSPRT(1300) : shield counts: -2 = no value.

HLOW(1300) : low voltage: -2 = no value.

HSGAL(1300) : solar gamma counts .3 - 9 MeV: -2 = no value.

HBGAL(1300) : background gamma counts .3 - 9 MeV: -2 = no value.

HSGAH(1300) : solar gamma counts > 9 MeV: -2 = no value.

HBGAH(1300) : background gamma counts > 9 MeV: -2 = no value.

HCUPS(1300) : solar cup charged particle count: -2 = no value.

HCUPB(1300) : background cup charged particle counts: -2 = no value.  
 HSLBS(1300) : solar slab charged particle counts: -2 = no value.  
 HSLBB(1300) : background slab charged particle counts: -2 = no value.  
 HGAIn(1300) : gain: 0 = no value 1 = high 2 = low.  
 HSTEP(1300) : step number: 0 = no value -2 = zero counts  
 HSCN(1300) : scan mode: 1 = full 2 = intermediate 3 = short  
 0 = no value.  
 IHORB(10) : array of orbit numbers.  
 IXHORB(10) : plot indices for the orbit numbers.  
 IDAY : day of year at start of plot frame.  
 LENPLT : value equals last valid plot index for plot frame.

C. Variables and arrays of interest either not in common storage or equivalenced to arrays in common.

HAMASK(47) : analog status masks (elements 1 - 32 unused).  
 HASBIT(47) : analog status indication bit value (elements 1 - 32 unused).  
 HASTEP(47) : analog value step =  $\frac{\text{analog readout} + 65}{45}$  (elements 1 - 32 unused).  
 HMFBIT(47) : main frame status bits (status bits propagated for half-scans in which the status word is found) (elements 1 - 27 unused).  
 IDTAPF : tape and file identifier for current file.  
 IXAPLT(30) :  $IXAPLT(I) = (UPTIME(1,I) - UPLTGO)/15.36$  = index of plot array for pane I.  
 IXASEQ(47) : analog subcom sequence identifiers (1 → sync word) (elements 1 - 27 unused).  
 IXBEGS : plot index for scheduled start of scan.  
 IXNEXS : plot index for scheduled start of scan following section.  
 IXP15 : plot index for frame in pane 15.  $IXP15 \equiv IXAPLT(15)$ .  
 JDSID(47) : digital subcom energy ID indicators (elements 1 - 32 unused).



JPXID(47) : plot indices in time sequence reconstruction  
(elements 1 - 32 unused).

JPXPAN(47) : pane indices in time sequence reconstruction  
0 → missing frame.

JXAPLT(15) : JXAPLT(1) = LXAPLT(2).

JXAPLT(J) = plot index for major frames containing  
the analog sync word in panes 15 - 29.

LXAPLT(1) is used to store the plot index for  
the most recent identifiable analog sync word  
from previously processed major frames.

KADDEN(30) : section addend parameters for statistics.

1. count incrementer for sections processed.
2. count incrementer for sections bypassed.
3. count incrementer for complete sections processed.
4. count incrementer for complete sections processed with  
no analog flags.
5. count incrementer for complete sections processed with  
no MF flags.
6. number of scheduled status words missing or not recog-  
nized (processed sections).
7. number of time gaps (imbedded within a Decomm file)  
- total.
8. number of time gaps (imbedded within a Decomm file)  
- total "short".
9. major frames lost due to time gaps.
10. major frames bypassed.
11. total major frames processed.
12. major frames processed with no flagged analog.
13. major frames processed with no flagged DSF.
14. major frames processed with no flagged MF.
15. total x-ray sequences selected for processing.
16. total x-ray sequences selected with no flags for digital  
subcom source data.

17. sync error instances count for frames processed.
18. sync error bit count for frames processed.
19. contradiction count (one message each).
20. day-night change count (of sections).
21. radioactive calibration count (of sections starting with radiation calibration in progress).
22. electronic calibration count (of sections starting with electronic calibration in progress).
23. total sections (processed or bypassed).
- 24 - 30. spares.

KFSUMY(40) : file summary information. KFSUMY(1) = HFSUMY(1):

	HFSUMY	KFSUMY
accumulators for section addends		1-30
I/O errors detected		31
begin time parameters (year, day of year, hour of day, minutes, seconds)	63-67	
end time parameters (year, day of year, hour of day, minutes, seconds)	68-72	
tape sequence count	73	} 37
file sequence count	74	
spares		38-40

KRSUMY : run summary information. KRSUMY(1) = HRSUMY(1):

	HRSUMY	KRSUMY
accumulators for section addends		1-30
I/O errors detected		31
earliest begin time parameters	63-67	
latest end time parameters	68-72	
final tape sequence count	73	
total files summarized	74	
files summarized but completely bypassed		38
spares		39-40

KTSUMY : tape summary information. KTSUMY(1) = HTSUMY(1):

	HTSUMY	KTSUMY
accumulators for section addends		1-30
I/O errors detected		31
earliest begin time parameters	63-67	
latest end time parameters	68-72	
tape sequence count	73	
final file count	74	
spares		38-40

KIOERR(2,3) : I/O error accumulators.  
KIOERR(1,J) = tape and file ID for J'th accumulator.  
KIOERR(2,J) = count of I/O errors.

LENHAF(4) : conversion table to get half-scan length from scan mode code.

LISPLT : number of elements in arrays used to store values to be plotted.

LXEND : GLB of pane indices after current section.

LXSYND : plot index for pane containing most recent "clear" low energy ID for digital subcom data.

LXSYNW : plot index for pane containing most recent "clear" sync word for wheel subcom data.

MAXGAP : maximum number of missing panes allowed without forcing plot output.

NPANES : count of panes in current section.

QNOPLT : .TRUE. if plots not wanted.

QSSCHG : .TRUE. implies a change in status exists at some major frame in the current section.

ULSYND : time (seconds of century) of major frame containing most recent "clear" sync word for energy ID for digital subcom data.

ULSYNW : time (seconds of century) of major frame containing most recent "clear" sync word for wheel subcom data.

## VI. Subroutine CALIB (X,NSUM)

Program Language : Fortran IV, G-Level  
 Date Written : August 1972  
 Programmer : S. Croteau  
 Installation : University of New Hampshire  
 Space Science Center  
 Computer : IBM-360-H50  
 Dummy Arguments : 1) X: an output array of three floating  
 point elements containing the three  
 peak channels for the spectrum.  
 2) NSUM: an output array of three full  
 word integer elements containing the sum  
 of counts surrounding each peak.  
 Subprograms Called : None  
 Description : This subroutine locates the two highest  
 peaks and ascertains that they are valid  
 peaks as opposed to bit errors. The pro-  
 gram then calculates an approximate guess  
 channel for the third peak. It then  
 searches for the third peak within a range  
 of the guess channel, -5 to the guess  
 channel, +5. The center of peak is then  
 calculated for the three peaks (results  
 are stored in X) and the counts are summed  
 for each of the three peaks (results are  
 stored in NSUM).  
 Program logic is as follows:  

$$E_1 = C_1 (n_1 + n_0)^2$$

1. Find  $n_1$  (highest counts/channel)
2. Find  $n_2$  (second highest counts/channel)

 Check if:
 

3.  $\Delta E = 1.33 - 1.17 = \text{Constant}$   
 $|n_2 - n_1| \leq 15$
4.  $n_1 < n_2$

5. Define:  $n_3' = \sqrt{\frac{2.525}{1.17/(n_1+n_0)^2}} - 80.2$

6. Find  $n_3$  (highest counts/channel)  
between  $n_3' - 5 \leq n_3 < n_3' + 5$

7. Define  $\overline{X(n_j)}$

$$\overline{n_j} = \frac{\sum_{i=-2}^{+2} N_{j+i} n_{j+i}}{\sum N_{j+i}} \quad j = 1, 2, 3$$

8. Define  $NSUM(N_j)$

$$N_j = \sum_{i=-2}^{+2} N_{j+i} \quad i = 1, 2, 3$$

# VII. Subroutine CALNDR (USOC,IYR,IMO,IDA,IHR,IMN,SEC)

Program Language	:	Fortran IV, G-Level
Date Written	:	February 1972
Programmer	:	S. Croteau
Installation	:	University of New Hampshire Space Science Center
Computer	:	IBM-360-H50
Dummy Arguments	:	<p>1) USOC: double precision input floating-point variable containing seconds of century.</p> <p>2) IYR: output integer full word containing year.</p> <p>3) IMO: output integer full word containing month.</p> <p>4) IDA: output integer full word containing day.</p> <p>5) IHR: output integer full word containing hour.</p> <p>6) IMN: output integer full word containing minutes.</p> <p>7) SEC: output floating point full word containing seconds.</p>
Subprograms Called	:	None
Description	:	<p>This subroutine accepts seconds of century as input via the argument list and converts this time to calendar time -- year, month, day, hours, minutes, and seconds. The input value in seconds of century is a 64-bit floating point word. The output variables are 32-bit integers except for the seconds which are stored in a 32-bit floating point word.</p>

### VIII. Subprogram FEEDER

Refer to Section 1 of this report for a description of this program.

## IX. Subroutine FORTIM (IRUNDA,IRUNT)

Program Language : IBM O/S 360 Assembler Language

Date Written : July, 1969

Programmer : A. Buck

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments : 1) IRUNDA: output integer full word  
containing the date.  
  
2) IRUNT: output integer full word  
containing the time of day in hundredths  
of a second.

Subprograms Called : None

Description : Subroutine FORTIM uses the time macro  
to fetch the date and the time from  
the computer clock. The date and the  
time are only as accurate as the cor-  
responding information entered by the  
operator.

Example : FORTIM is called at 2.37 seconds after  
3 minutes after 1 o'clock, A.M. on  
February 10, 1970. When control is  
returned from subroutine FORTIM, IRUNDA  
contains 70041 in binary form and IRUNT  
contains 3780237 in binary form.



## X. Subroutine FPUT (IDSN,ICMD,IRSLT,IOVEC)

Program Language : IBM O/S 360 Assembler Language

Date Written : February 4, 1971

Programmer : A. Buck

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments : 1) IDSN: a full word integer input parameter specifying the data set to which activity is directed. The integer stored at IDSN must be greater than zero but less than 21.

2) ICMD: a full word input integer variable containing a request type code which specifies the activity to be performed.

a) ICMD = 1 implies that a logical record is to be written.

b) ICMD = 2 implies the data set is to be ended. Once ended, the data set is unavailable to the calling program through the same data set number.

3) IRSLT: a full word output integer variable in which subroutine FPUT can place a return code as follows:

a) IRSLT = -1 implies request was illegal for one of the reasons below:

i. IDSN contained a value less than 1 or greater than 20.

ii. ICMD contained a value less than 1 or greater than 2.

iii. A write request was made to a data set that had been ended.

iv. An endfile request was made to a data set either before the first write request or after the data set had been ended.

b) IRSLT = 0 implies request was executed normally.

c) IRSLT = 1 implies an I/O error occurred while writing a block. The data set was ended.

4) IOVEC: an input vector whose high order byte is the first byte to be written (when ICMD = 1) in a logical record.

The number of bytes written is specified by the LRECL subparameter on the DD statement describing the data set.

Subprograms Called : None

Description : Subroutine FPUT is a QSAM output subroutine to write logical records from an area defined by the calling program. This routine is specifically designed to allow a Fortran program to create without format control an extensive data set to which the DCB subparameter 'RECFM = FB' applies. For data sets to which 'RECFM = F' or RECFM = U' applies, subroutine FTWRIT may be more easily used.

Subroutine FPUT can manage as many as 20 data sets in the same job step. As many as desired may be active simultaneously. Once the endfile command has been issued to a data set, subroutine FPUT will refuse to perform any further activity for that particular data set. The large number of data sets allows writing as many as 20 files to the same magnetic tape volume as a possible application.

DD Statements required...

If subroutine FPUT is called with IDSN=1 (or 2, ..., or 20) the user must supply a DD Statement having the DD name FPUT01 (or FPUT02, ..., or FPUT20). The DD Statement must include the DCB subparameters 'RECFM', 'LRECL', and 'BLKSIZE' but must omit 'DSORG' and 'MACRF'.

## XI. Subroutine FTWRIT (IDSN, ICMD, IEXT, IRSLT, OUTBLK)

Program Language : IBM O/S 360 Assembler Language

Date Written : December 31, 1970

Programmer : A. Buck

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments : 1) IDSN: a full word input integer  
variable specifying the number (1,2,  
or 3) of the data set to which the  
command is directed.

2) ICMD: a full word input integer  
variable containing a command code  
specifying the action to be taken by  
the subroutine as follows...

a) ICMD = 1 : write a block to  
the data set.

b) ICMD = 2 : write a tape mark  
and position the tape past the  
tape mark.

c) ICMD = 3 : close the data set.  
(endfile).

3) IEXT: a full word input integer  
variable containing the number of  
bytes to be written when ICMD = 1.  
(If the value of ICMD is not 1, the  
value of IEXT is ignored.)

4) IRSLT: a full word output integer  
variable into which subroutine FTWRIT  
can place a response code. When con-  
trol is returned to the calling pro-  
gram the meaning of the contents of  
IRSLT is as follows...

a) IRSLT = -1 : Command was invalid.  
One of the following occurred.

i. IDSN was not 1, 2, or 3.

ii. ICMD was not 1, 2, or 3.

iii. A request to process the  
data set was made after the  
data set had been closed.

b) IRSLT = 0 : Command executed  
successfully.

c) IRSLT = 1 : An uncorrectable  
I/O error occurred while writing  
the block. The data set was closed.

5) OUTBLK: any variable or array element  
(integer, real, logical or complex) whose  
high order byte will be the first byte  
written to the block when ICMD = 1.

Subprograms Called : None

Description : This subroutine is a BSAM output routine  
to write data sets to which the DCB sub-  
parameter 'RECFM = U' applies. As many  
as 3 such data sets may be open at one  
time. The subroutine...

1) Writes blocks stored in the  
calling program.

2) On command, will write a tape  
mark and position the tape beyond  
the tape mark. This enables the  
calling program to write more than  
one file on an unlabelled tape with-  
out additional DD Statements.

3) On command, will close the data  
set. Once closed, the subroutine  
will consider invalid all subse-  
quent requests to process the data  
set.

4) Returns a response code to the  
calling program to indicate the  
success or failure of the subroutine  
to execute the command.

### DD Statements required...

A DD Statement having the name FTWT01 (or FTWT02 or FTWT03) must be supplied if subroutine FTWRIT is called with IDSN = 1 (or 2 or 3). The only DCB subparameters needed on the DD Statement are 'BLKSIZE' and, where appropriate, 'DEN' and 'TRTCH'. The label subparameter 'LABEL = (,,OUT)' must not be used for any data set to which the write-tape-mark command (ICMD = 2) is directed.

## XII. Subroutine MOVER (AREAL, IDISP1, AREA2, IDISP2, LEN)

Program Language : IBM O/S 360 Assembler Language

Date Written : March 23, 1971

Programmer : A. Buck

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments :

- 1) AREAL: any variable or array element (integer, real, logical, or complex) whose highest order byte is used as the 'output vector reference byte'.
- 2) IDISP1: a full word input integer whose value is the number of bytes to be skipped, starting with the 'output vector reference byte' to locate the highest order byte of the first (output) area.
- 3) AREA2: any variable or array element (integer, real, logical, or complex) whose highest order byte is used as the 'input vector reference byte'.
- 4) IDISP2: a full word input integer whose value is the number of bytes to be skipped, starting with the 'input vector reference byte' to locate the highest order of the second (source) area.
- 5) LEN: a full word input integer whose value is the number of bytes to be moved.

Subprograms Called : None

Description : Subroutine MOVER causes the contents of a first area of core storage to be replaced by the contents of a second area of same length as the first. Replacement occurs on one byte at a time starting at the highest order bytes of the two areas and proceeding through progressively lower order bytes. Any number of bytes may be moved by one call to subroutine MOVER.

Programming notes...

The two areas may overlap in any way.  
Suppose the Fortran calling program  
has dimensioned a vector by the state-  
ment

Dimension A (1000)

Examples are shown below in the use of  
subroutine MOVER to 'Propagate' or to  
'Downshift' values in the vector.

Propagation...

The two statements

A(1) = 0.0

CALL MOVER (A,4,A,0,3996)

Could be used in place of the loop

Do 1 I = 1,1000

1 A(I) = 0.0

Downshifting...

The statement

CALL MOVER (A,0,A,8,3992)

Could be used in place of the loop

Do 2 I = 1,998

2 A(I) = A(I+2)

## XIII. Subroutine OUTPUT (version C)

Program Language : Fortran IV, H-Level  
Date Written : January 1972  
Programmer : S. Croteau  
Installation : University of New Hampshire  
Space Science Center  
Computer : IBM-360-H50  
Dummy Arguments : None  
Subprograms Called : FEEDER, MOVER, CALIB, CALNDR, FPUT

The purpose of this program is to output the experiment data in any of six different forms. The options desired are specified by a control card read by the main program. The options are:

- A. Engineering plot
- B. Line summary
- C. Calibration cards
- D. X-ray tape
- E. Good time cards
- F. Main frame data tape

Any combination of options may be selected for one production run.

Subroutine OUTPUT receives the input data from the main data processing program and from FEEDER through common storage. When called by the main program, OUTPUT expects a section of coherent data to be stored in panes 15 - 26 of the DECISION



window. If possible, this should cover three minutes of accumulated data and begin with a background scan. Each pane of the decision window contains one major frame of data.

There is a program section in OUTPUT for each different option. It is, in fact, six programs in one. The following is a description of each of these program sections:

#### A. Engineering plot section:

The plotting section of OUTPUT creates tables that are 1300 half-words long corresponding to the 1300 elements in each engineering plot frame (13.0 in/.01 in). OUTPUT processes up to 12 major frames per call. This means that up to 12 elements of each tables are filled with one call to OUTPUT. The main data processing program sets up plot indices to be used by OUTPUT. After the tables have been filled, they are stored on disk by the main program and the plot index is reset to start a new plot frame. The tables for each engineering plot frame are stored on disk until the end of the run. In the second job step, the disk is read by the plot production program and the plotter tapes are generated.

#### B. Line summary section:

The line summary section prints out two types of lines; the status line and the summary line. The status line is an alphanumeric representation of status, shield and central detector on-off, and the step number. This line is printed whenever one of these values changes. An asterisk precedes the changing value. This line is also printed at the start of each new file and after a long time gap. The summary line is a summary of analog and status information for three minutes of data or one full scan, three intermediate scans, or six short scans. The time printed is the beginning of accumulation time. The status word printed with the line is the status that was read out with the data and is not the status at accumulation. The message number printed with a summary line associates that line with an error message in the error listing.

#### C. Calibration card section:

This section interrogates HDCAL(1) to see if the main frame data was accumulated in the calibrate mode. If so, FEEDER is called to provide the main frame data in partitions 1 - 12 of the SCAN window. Subroutine CALIB is then called

to identify the three peaks, calculate the integrated peak channels, and sum the counts surrounding each peak. The peak channels and counts are returned and punched out along with the time at midpoint of accumulation.

#### D. X-ray section:

The x-ray data is provided to this program through the DECISION window (i.e. DECIDE common area) in two forms; source values as they are on the Decomm tape or x-ray counts corrected for scaling. The values that are used to create the x-ray tape in this program are the corrected or interpreted values. The digital subcom data for one cycle is contained in three major frames. The third major frame for one cycle is the first major frame for the next cycle.

Major frame #1 contains:

DSM WORD #		
9	60 - 120 keV	Data - Background (for previous cycle)
19		Energy ID (Low) all zeros
22		Quadrant time - Solar
23	7.5 - 15 keV	Data - Solar
24		Quadrant time - Background
29	7.5 - 15 keV	Data - Background
47	15 - 30 keV	Data - Solar

Major frame #2 contains:

DSM WORD #		
9	15 - 30 keV	Data - Background
19		Energy ID (High) all ones
22		Quadrant time - Solar
23	30 - 60 keV	Data - Solar
24		Quadrant time - Background
29	30 - 60 keV	Data - Background
47	60 - 120 keV	Data - Solar

Major frame #3 contains:

DSM WORD #		
9	60 - 120 keV	Data - Background
19		Energy ID (Low) all zeros
22		Quadrant time - Solar
23	7.5 - 15 keV	Data - Solar
24		Quadrant time - Background
29	7.5 - 15 keV	Data - Background
47	15 - 30 keV	Data - Solar

This program section creates one x-ray tape record for each x-ray cycle. A fill of -1's is used to represent missing frames.

#### E. Good time card section:

The good time card section is responsible for making the decision as to whether the data in a spectrum is to be considered good or whether it is to be bypassed. The criteria used for determining data to be bypassed are as follows:

1. Central detector off
2. Time gap (interfile gap).
3. Any information missing - sync word loss, major frame loss.

The program checks LASHV (last high voltage) and NOWHV (now high voltage) to decide whether or not the spectrum meets the requirement of central detector on.

If the last high voltage and the present high voltage are on, the spectrum is accepted. If neither is on, the spectrum is rejected. If the last high voltage is on and the present high voltage is off, the spectrum is accepted because it was accumulated when high voltage was on. If the last high voltage is off and the present high voltage is on, the spectrum is rejected because it was accumulated during high voltage off.

This program section saves the begin time and the end time (midpoint of accumulation times) in seconds of century in a double precision (64-bit) floating point word. It also saves whether the begin time starts at a solar quadrant or includes both quadrants and whether the end time ends at a background quadrant or includes both quadrants. When a rejection requirement is fulfilled, a good time card is punched and new begin and end times are saved.

#### F. Main frame data tape section:

This program section begins by determining whether the data section provided by the data processing program is complete, that is, whether all major frames are present for each quadrant. If so, FEEDER is called to fetch the main frame data for twelve major frames. If not, the program initializes the scan window to -1's and calls FEEDER to fetch the main frame data for each major frame present in the three minute span, placing the main frame data in the appropriate partition allowing for major

frame loss. This means that 12 partitions of the scan window are filled, either with main frame data or -1's to represent data loss.

The same logic is followed to fill in the analog word array for each quadrant. A fill of -1's is used to represent missing information. A flag is set to indicate the presence of a time gap and a count of major frames present in the quadrant is given. The real time is converted from seconds of century to calendar time and introduces the record on tape.

OUTPUT employs the same labelled common areas as described in Part V of this section. In addition, there is a labelled common area called /SMOLST/ used to transfer information to subroutine CALIB and containing the following arrays:

- SKAN(377,2) : this array is included because subroutine CALIB was written for use by another program and the labelled common area requires this array.
- SSCAN(377,2) : this array is also included to make this program compatible with the storage requirements of CALIB.
- HARRAY(768) : the second half of this array contains the main frame data for the solar quadrant of the calibration.

Here is an alphabetic list of variables and arrays of interest not in common followed by the description for each term mentioned:

ASOD	IANLG	IPRST	IVD	UHOURL(24)
AI(3)	IANLT	IQUAD(2)	IXPLT	UPT
BCUP	IASTER(2)	IREC	IXPLT1	USCAN
BGLOG	ICHGPN(12)	IRSLT	JPART	USODAY
BGSUM	ICHIV	ISCAN(4)	LMONTH(12)	UTME
BSLAB	ICHV	ISCNH	MONTH(12)	
CKTMB	ICPHV	ISHHV	NSS(3)	
CKTMS	ICPVD	ISKN	PANE	
CLOCK	IDANI(2)	ISODAY	QCAL	
COUNT	IDETT	ISOFF	SCAN	
DDTMB	IDHLD	ISTAT1	SCANTM(3)	
DDTMS	IELTP	ISTAT2	SCUP	
DIVB	IEPD	ISTAT3	SCUPB1	
DIVS	IGHLD	ISTAT4	SCUPS1	
DTEMP	IHILO(2)	ISTAT5	SGAMB	
DTMB	ILETT(13)	ISTAT6	SGAMS	
DTMS	ILVD	ISTAT7	SGLOG	
ETEMP	ILVPS	ISTPHD	SGSUM	
HCLOCK(64)	IMAG	ISYNER	SSLAB	
HFULL(768)	IMGY(2)	ISYN1	SSLABB	
HHOLD	IONOF(3)	ISYN2	SSLABS	
HSTPNO	IPRINT	ITD	STPNO	

ASOD : floating point full word variable containing the seconds of day for the major frame being processed (engineering plot).

Al(3) : floating point full word array containing the integrated peak channels for a radiation calibration after a call to CALIB (calibration card).

BCUP : floating point full word containing the average background cup for the data section being processed (line summary).

BGLOG : floating point full word containing the average background gamma x .1 (line summary).

BGSUM : floating point full word containing the quotient of the background gamma sum divided by background live time (line summary).

BSLAB : floating point full word containing the average background slab for the data section being processed (line summary).

CKTMB : double word variable containing the background clock time (= PHA clock time x 8 x 1/390.625) (line summary).

CKTMS : double word variable containing the solar clock time (= PHA clock time x 8 x 1/390.625) (line summary).

CLOCK : floating point full word containing the PHA clock time (line summary).

COUNT : floating point full word variable containing the step number in raw form (engineering plot and line summary).

DDTMB : double word variable containing the background dead time (= DTMB x .001515) (line summary).

DDTMS : double word variable containing the solar dead time (= DTMS x .001515) (line summary).

DIVB : floating point full word containing the background live time (line summary).

DIVS : floating point full word containing the solar live time (line summary).

DTEMP : floating point full word containing the average detector temperature divided by 10 (line summary).

DTMB : double word variable containing the sum of the value of the 2 lowest order bits of the status word for a background quadrant and 4 x the value of the 60th main frame experiment word in the last major frame of that quadrant (line summary).

DTMS : double word variable containing the sum of the value of the 2 lowest order bits of the status word for a solar quadrant and 4 x the value of the 60th main frame experiment word in the last major frame of that quadrant (line summary).

ETEMP : floating point full word containing the average electronic temperature divided by 10 (line summary).

HCLOCK(64) : integer half-word array equivalenced to partition 1 of the scan window into which the main frame data of the last major frame of each quadrant is placed to obtain the PHA clock time in HCLOCK(64) (line summary).

HFULL(768) : integer half-word array equivalenced to the first twelve partitions of the scan window - may contain all the main frame data for one full scan (main frame data tape).

HHOLD : integer half-word set to -1 to determine whether the background quadrant is missing or whether a time gap exists in the quadrant (main frame data tape).

HSTPNO : half-word integer variable containing the fixed point value for the step number (line summary).

IANLG : integer full word variable used as an index for analog sub-subcombed words ASC 1:1-6, ASC 2:1-6 (engineering plot and line summary).

IANLT : integer full word variable used to save the normal - alternate quadrant status (engineering plot).

IASTER(2) : full word array containing the following alphabetic information: blank and \*.

ICHGPN(12) : full word integer array containing the pane numbers of sync major frames where status change occurs (not including background - solar change).

ICHIV : integer full word variable containing the average central detector HVPS analog (line summary).

ICHV : integer full word variable containing the sum of the central HVPS analog for the data section being processed (line summary).

ICPHV : integer full word variable containing the sum of the charged particle HVPS analogs for the data section being processed (line summary).

ICPVD : integer full word variable holding the number of major frames used in the charged-particle HVPS analog summation (line summary).

IDANI(2) : full word array containing alphameric DAY or NITE.

IDETT : integer full word variable containing the sum of the detector temperature for the data section being processed (line summary).

IDHLD : integer full word variable used to save the day - night status in the engineering plot section.

IELTP : integer full word variable containing the sum of electronic temperature for the data section being processed (line summary).

IEPD : integer full word variable holding the number of major frames used in the electronic temperature summation (line summary).

IGHLD : integer full word variable used to save the gain status in the engineering plot section.

IHILO(2) : full word array containing alphameric HI and LO.

ILETT(13) : full word array containing alphameric F,I,S,A,N, E,blank,R,D,M,G,P,C.

ILVD : integer full word variable holding the number of entries in the low HVPS analog summation.

ILVPS : integer full word variable containing the average of the LVPS analogs for the data section being processed (line summary).

IMAG : integer full word variable used to save the magnetometer - gyro reference in the engineering plot section.

IMGY(2) : full word array containing alphameric MAG and GYRO.

IONOF(3) : full word array containing the following alphameric information: ON, OFF, and blank.

IPRINT : integer full word variable set to 1 to indicate that a status line must be printed due to a parameter change (line summary).

IPRST : integer full word variable used as a print-status-line indicator (line summary).

IQUAD(2) : full word array containing alphameric N and A.

IREC : integer full word variable used as a record counter for x-ray records written (x-ray tape section).

IRSLT : integer full word variable containing the return code as a result of a call to FPUT (x-ray tape and main frame data tape sections).

ISCAN(4) : double word array containing the following alphameric information: FULL, INTER, SHORT, INVLD.

ISCNH : integer full word variable used to save the scan mode in the engineering plot section.

ISHHV : integer full word variable containing the average shield HVPS analog for the data section being processed (line summary).

ISKN : full word variable containing the alphameric representation of the scan mode (F = full, I = intermediate, S = short).

ISODAY : integer full word variable containing seconds of day for the major frame being processed (line summary).

ISOFF : integer full word variable used to save shield on - off (engineering plot).

ISTAT1 : full word variable containing the alphameric representation for normal - alternate quadrant (N = normal, A = alternate) (line summary).

ISTAT2 : full word variable containing the alphameric representation for electronic calibration on - off (E = on, blank = off) (line summary).

ISTAT3 : full word variable containing the alphameric representation for radiation calibration on - off (R = on, blank = off) (line summary).

ISTAT4 : full word variable containing the alphameric representation for day - night indicator (D = day, N = night) (line summary).

ISTAT5 : full word variable containing the alphameric representation for magnetometer - gyro reference (M = magnetometer, G = gyro) (line summary).



ISTAT6 : full word variable containing the alphameric representation for charged particle on - off (P = on, blank = off) (line summary).

ISTAT7 : full word variable containing the alphameric representation for central detector on - off (C = on, blank = off) (line summary).

ISTPHD : integer full word variable used to save the step number in the engineering plot section.

ISYNER : integer full word variable equal to ISYN2 (line summary).

ISYN1 : integer full word variable containing the count of non-fill minor frames with sync errors in the data section being processed (line summary).

ISYN2 : integer full word variable containing the total sum of bits in error in non-fill minor frame sync for the data section being processed (line summary).

ITD : integer full word variable holding the number of major frames used in the detector temperature summation (line summary).

IVD : integer full word variable holding the number of major frames used in the central HVPS analog summation (line summary).

IXPLT : integer full word variable containing the plot index of the major frame being processed (engineering plot).

IXPLT1 : integer full word variable used as plot index through a time gap (engineering plot).

JPART : integer full word variable containing the number of the first partition where main frame data is to be placed in the call to FEEDER (main frame data section).

LMONTH(12) : integer full word array containing the day of year of the end of each month of a leap year.

MONTH(12) : integer full word array containing the day of year at the end of each month of an ordinary year.

NSS(3) : integer full word array containing the sum of counts surrounding the three calibration peaks after a call to CALIB.

PANE : integer full word variable containing the pane number of the major frame being processed (from IXPANE to LXPANE).

QCAL : = .TRUE. if a call to CALIB must be made and a calibration card punched (calibration card section).

SCAN : floating point variable containing the number of seconds to subtract from the background read-out time in order to obtain the time at beginning of data accumulation. (184.32 for full scan, 61.44 for intermediate scan, 30.72 for short scan).

SCANTM(3) : floating point array containing the number of seconds to subtract from background read-out time to obtain the time at midpoint of data accumulation (92.16 for full scan, 30.72 for intermediate scan, 15.36 for short scan).

SCUP : floating point full word variable containing the average solar cup for the data section being processed (line summary).

SCUPB1 : floating point full word variable containing the sum of the sixth ASC word from each major frame of the data section being processed (charged particle cup, background) (line summary).

SCUPS1 : floating point full word variable containing the sum of the tenth ASC word from each major frame of the data section being processed (charged-particle cup, solar) (line summary).

SGAMB : floating point full word variable used to contain the sum of the third ASC word from each major frame in the data section being processed (gamma .3 - 9.1 MeV, background) (line summary).

SGAMS : floating point full word variable used to contain the sum of the seventh ASC word from each major frame in the data section being processed (gamma .3 - 9.1 MeV, solar) (line summary).

SGLOG : floating point full word variable containing the average solar gamma x .1 (line summary).

SGSUM : floating point full word variable containing the quotient of the solar gamma sum divided by the solar live time (line summary).

SSLAB : floating point full word variable containing the average solar slab for the data section being processed (line summary).

SSLABB : floating point full word variable containing the sum of the fifth ASC word from each major frame of the data section being processed (charged particle slab, background) (line summary).

SSLABS : floating point full word variable containing the sum of the ninth ASC word from each major frame of the data section being processed (charged-particle slab, solar) (line summary).

STPNO : floating point full word variable used to calculate the step number (engineering plot and line summary).

UHOURL (24) : double word array containing seconds of hour for hours 1 to 24.

UPT : double word variable used to hold time in seconds of century.

USCAN : double word variable containing the number of seconds to subtract from read-out time to obtain midpoint of accumulation time.

USODAY : double word variable used to contain the seconds of day for the major frame being processed.

UTME : double word variable containing the time at midpoint of accumulation in seconds of century for a calibration (calibration card section).

## XIV. Subroutine PLOUT

Program Language : Fortran IV, G-Level  
Date Written : January 1972  
Programmer : A. Buck  
Installation : University of New Hampshire  
Space Science Center  
Computer : IBM-360-H50  
Dummy Arguments : None  
Subprograms Called : FTWRIT, MOVER  
Description : Subroutine PLOUT writes the tables  
created by the plot section of subrou-  
tine OUTPUT onto disk. The tables con-  
tain the information necessary for one  
plot frame. The main program of Pass-I  
initializes these tables.

## XV. Subroutine PUZZLE (version C)

Program Language : Fortran IV, G-Level

Date Written : January 1972

Programmer : A. Buck

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments : None

Subprograms Called : MOVER

Description : This program tries to solve 'puzzling'  
situations by realigning major frames  
surrounding sync word loss or time gap  
occurrence. Certain messages are printed  
out depending on the problem-solving  
method used.

## XVI. Function QUASH (HAR,HMASK,LEN)

Program Language : IBM O/S 360 Assembler Language

Date Written : January 1972

Programmer : A. Buck

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments : 1) HAR: flag word  
2) HMASK: mask  
3) LEN: integer full word specifying  
the length in bytes of the flag word  
and the mask.

Subprograms Called : None

Description : QUASH turns off the flag bits you are  
willing to ignore and then gives a true-  
false reading.

## XVII. Subroutine SUMARY (KFSUMY,HFSUMY,KNFDSN,KAFDSN,KJFDSN,I)

Program Language : Fortran IV, G-Level  
 Date Written : January 1972, Revised March 14, 1972  
 Programmer : A. Buck  
 Installation : University of New Hampshire  
 Space Science Center  
 Computer : IBM-360-H50  
 Dummy Arguments : 1) KFSUMY: an input vector containing  
 the accumulated statistics (full word  
 integers) (see file summary descrip-  
 tion in Part V of this section).  
 2) HFSUMY: the same input vector.  
 (half-word integers)  
 3) KNFDSN: an input integer variable  
 defining the first data set number on  
 which the display is to be written.  
 4) KAFDSN: an input integer variable  
 defining the last data set number on  
 which the display is to be written.  
 5) KJFDSN: an input integer variable  
 containing the greater of [1,(KAFDSN-  
 KNFDSN)].  
 6) I: an input integer variable defining  
 the type of statistical display required.  
 Subprograms Called : None  
 Description : This program assists the main program  
 by creating and printing a display of  
 statistics accumulated in an array  
 passed by the main program. Subroutine  
 SUMARY is responsible for accumulating  
 time intervals spanned by individual  
 tapes (when I = 2) for use in the final  
 run summary report (I = 3)

## XVIII. Subroutine SUMCAL (KFSUMY,KTSUMY)

Program Language : IBM O/S 360 Assembler Language

Date Written : January 1972

Programmer : A. Buck

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments : 1) KFSUMY: an input vector containing  
the accumulated statistics for a file.

2) KTSUMY: an output vector containing  
the accumulated statistics for the tape.  
(see summary descriptions in Part V of  
this section).

Subprograms Called : None

Description : This program compares the begin time  
of the file with the begin time in the  
tape accumulator. If the file time is  
earlier, the earlier time is moved to  
the tape summary begin time. The end  
times are also compared and the later  
time is moved to the tape summary  
end time.



## XIX. Plot Production Program (OSOP104A)

Program Language : Fortran IV, G-Level  
Date Written : February 1972  
Programmer : A. Buck  
Installation : University of New Hampshire  
Space Science Center  
Computer : IBM-360-H50  
Subprograms Called : FGET,INITPL,OUTLNE,ENPLTA,PLOT (see Section 7)

The purpose of this program is the distribution of plots on the plotter tapes and plot construction. The program reads the disk file containing the plot frame count. From this count, it computes the number of tapes that will be required to plot a maximum of three plot frames per tape. A maximum of three plot frames per tape was selected because of the poor condition of the Calcomp plot hardware at the University of New Hampshire Computation Center. After two hours of plotting, the hardware has a tendency to malfunction. It usually takes about two hours to plot three frames. There is no restart option to begin plotting again at the frame in which the malfunction occurred.

The plot production program reads the engineering plot tables stored on disk and calls INITPL,OUTLNE, and ENPLTA for actual plot construction. For one week of data, seven plotter tapes are generated containing three plot frames per tape.

## XX. Subroutine ENPLTA

Program Language : Fortran IV, G-Level  
Date Written : October 1971  
Programmer : S. Croteau  
Installation : University of New Hampshire  
Space Science Center  
Computer : IBM-360-H50  
Dummy Arguments : None  
Subprograms Called : PLOT (see Section 7), CHAR (see Section 7),  
OUTLINE  
Description : This program plots one plot frame consisting of 23 arrays dimensioned at 1300 half words each. Each half word represents one major frame of data and is represented by one hundredth of an inch on the "X" axis. Each array represents an individual plot within the total engineering plot frame.

ENPLTA reads and plots each table one at a time. The ORBIT number printed is the ascending-node-crossing number. The day of year for each frame is also written on the disk record.

## XXI. Subroutine FGET (IDSN,IRSLT,IOVEC)

Program Language : IBM O/S 360 Assembler Language

Date Written : February 4, 1971

Programmer : A. Buck

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments : 1) IDSN: a full word input integer containing the data set number to which the call is directed. (IDSN must be 1, 2, or 3).

2) IRSLT: a full word output integer in which subroutine FGET may place an activity-result code as follows...

a) IRSLT = -1 if the requested activity was invalid. This code is returned if either of the following occur:

i. IDSN contains a number less than 1 or greater than 3.

ii. Data set has been previously closed.

b) IRSLT = 0 if data is read and no I/O error is detected.

c) IRSLT = 1 if an I/O error occurred on the block containing the data placed in IOVEC. The calling program has the responsibility for accepting or rejecting the data. In addition, the calling program is responsible for proper reaction to the remaining records in the block for which IRSLT will be 0.

d) IRSLT = 2 if end-of-file was detected. The contents of IOVEC are not changed. The data set is closed and cannot be used again by subroutine FGET with the same value of IDSN.

3) IOVEC: an output vector in which FGET can place the record. The actual variable used for IOVEC may be integer, real, logical, or complex. The first leftmost byte of the data record is placed in the high-order byte of IOVEC when reading occurs. The number of bytes placed in IOVEC is specified by the LRECL subparameter of the DD Statement.

Subprograms Called : None

Description : This subroutine is a QSAM input subroutine to read logical records into an area defined by the calling program. This routine is specifically designed to supply a Fortran calling program with logical records from a data set to which the DCB subparameter 'RECFM = FB' applies. For data sets to which 'RECFM = F' or 'RECFM = U' applies, subroutine FTREAD may be more easily used. 'RECFM = V' is specifically prohibited.

Subroutine FGET can manage as many as 3 data sets during the same job step. The data sets may be open (active) simultaneously provided no two data sets are on the same magnetic tape.

DD Statements required...

If FGET is to be called with IDSN = 1 (or 2 or 3), then the user must describe the data set by a DD Statement having the DD name FGET01 (or FGET02 or FGET03). The DD Statement ~~must~~ include the DCB subparameters 'RECFM', 'LRECL', and 'BLKSIZE' but must omit 'DSORG' or 'MACRF'

## XXII. Subroutine INITPL

Program Language : Fortran IV, G-Level

Date Written : September 1971

Programmer : S. Croteau

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments : None

Subprograms Called : PLOT

Description : This subroutine makes the initial call to PLOT (see Section 7) in preparation for the engineering plot. The call to this subroutine is made by the main program in the second step of PASS-I. The subroutine requests that PLOT calculate the scale constants but neither a plotting area outline nor grid is drawn. The physical size of the engineering plot is 13.0 inches in the "x" direction and 10.0 inches in the "y" direction.

## XXIII. Subroutine OUTLNE

Program Language : Fortran IV, G-Level

Date Written : September 1971

Programmer : S. Croteau

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments : None

Subprograms Called : PLOT,CHAR

Description : This program plots the frame and labeling  
information for the engineering plot using  
the PLOT and CHAR software provided by  
the Computation Center (see Section 7).  
The call to this subroutine is made by  
the main program in the second step of  
PASS-I each time a new frame is desired.



IPLOT      1 = Plot.  
            0 = No plot.

ILIST      1 = Line summary.  
            0 = No line summary.

ICALIB     1 = Main frame display of calibration.  
            0 = No display.

IXRAY      1 = Create X-ray tape.  
            0 = No tape.

IGDT      1 = Punch good time cards.  
            0 = No good time cards.

IMFD      1 = Create MF data tape.  
            0 = No MF data tape.

HMAXTA     Maximum number of tapes to be processed.

HMAXFI     Maximum number of files to be processed.



## B. Experimenter Data Tape

This tape format was sent to EXP3 by Mr. Henry Linder, GSFC.

### 1. Data Format for OSO-H Experimenter Tape University of New Hampshire Experimenter Number 3

<u>Characters</u>	<u>Contents</u>
1 - 12	Time field.
13 - 18	Spin rate (rad/sec).
19 - 24	Pitch angle (rad).
25 - 30	Roll angle (rad).
31 - 36	Aspect angle (rad).
37 - 42	Right ascension of spin axis (rad).
43 - 48	Declination of spin axis (rad).
49 - 54	Right ascension of roll axis (rad).
55 - 60	Declination of roll axis (rad).
61 - 66	Right ascension of aspect axis (rad).
67 - 72	Declination of aspect axis (rad).
73 - 78	Azimuth time of encoder pulse with respect to beginning of minor frame 1.
79 - 84	Attitude flag field. See Attachment IV.
85 - 90	Spare.
91 - 168	Same type of data as in characters 13 - 90 but for time $T_2 = T_1 + 15.36$ sec (i.e., for the second major frame) one major frame contains 48 minor frames.
169-170+12N	Minor frame word 11.
171-172+12N	Minor frame word 12.
173-174+12N	Wheel subcommutator.
175-176+12N	Sail subcommutator.
177-178+12N	Digital subcommutator.
170-180+12N	Flag field for each minor frame.

### 2. Explanations about the Format

Where  $0 < N < 95$  corresponds to main frames 1 - 48 containing a complete subcom cycle followed by main frames 1 - 48 containing the next complete subcom cycle; thus every record length is 1320 6-bit characters or 220 36-bit computer words.

The tapes will be written with odd parity and a density of 556 BPI. Each file will be started with a label record, as described in Attachment I, followed by records filled with ORBIT information as described

in Attachment V, and then will be followed by the PCM data records. An end-of-file (EOF) mark closes the file. Several files are written on the same tape until two EOF marks indicate the end of data on the tape.

The time field contains subcom slip flag, day count of year, and milliseconds of day. The time field is described in Attachment II.

The attitude data is referenced to the time  $T_1$  given in the time field except for the second set of attitude data which is referenced to time  $T_2 = T_1 + 15.36$  sec. The attitude flag field is described in Attachment IV. All attitude words are written in IBM-7094 floating point format.

The flag field contains the frame number (1-48) in the subcom cycle, number of bit errors in the main frame sync word, fill data flag, and day/night indicator. The flag field is described in Attachment III.

Any changes to the above-described data tape format should be directed to Henry G. Linder, GSFC, Code 565, Greenbelt, Maryland, 20771.

ATTACHMENT ILabel Record Description

<u>Characters</u>	<u>Form</u>	<u>Contents</u>
1 - 5	BCD	Satellite identification number.
6 - 7	BCD	Year.
8 - 10	BCD	Station number.
11 - 14	BCD	Station tape number.
15 - 16	BCD	Analog file number.
17 - 18	BCD	Buffer file number.
19 - 21	BCD	Day count of year.
26 - 30	BCD	Start time (sec).
32 - 36	BCD	End time (sec).
37 - 38	BCD	Experimenter number.
39 - 40	BCD	Decomm number.
41 - 42	BCD	Reel number.
43 - 44	BCD	Master tape file.
45 - 48	BCD	Master tape number.
49	BCD	Type of attitude computation: 0 = based on star sensor inputs at night and SORE inputs in day time, 1 = based on magnetic field measurement inputs and SORE inputs in day time (accuracy 2-3).
50 - 60	BCD	Fill data (blank).
61 - 66	Floating Pt.	Day count of year.
67 - 72	Floating Pt.	Seconds of day.
73 - 78	Floating Pt.	Semi-Major axis, $a$ (km).
79 - 84	Floating Pt.	Eccentricity, $e$ (ratio).
85 - 90	Floating Pt.	Inclination, $I$ (deg).
91 - 96	Floating Pt.	Right ascension of ascending node, $\Omega$ (deg).
97 - 102	Floating Pt.	Rate of change of R.A. of ascending node (deg/day).
103 - 108	Floating Pt.	Argument of perigee, $\omega$ (deg).
109 - 114	Floating Pt.	Rate of change of argument of perigee (deg/day).
115 - 120	Floating Pt.	Period, $P$ (min).
121 - 126	Floating Pt.	Rate of change of period, $\dot{P}$ (deg/day).
127 - 132	Floating Pt.	Date.
133 - 138	Floating Pt.	Day count of year coordinate system.

ATTACHMENT I

<u>Characters</u>	<u>Form</u>	<u>Contents</u>
139 - 144	Floating Pt.	Apparent sidereal time in radians.
145 - 150	Floating Pt.	No. of frames expected.
151 - 156	Floating Pt.	No. of frames recovered.
157 - 162	Floating Pt.	No. of frames recovered.
163 - 168	Floating Pt.	No. of frames with 0 bit errors.
169 - 174	Floating Pt.	No. of frames with 1 bit errors.
175 - 180	Floating Pt.	No. of frames with 2 bit errors.
181 - 186	Floating Pt.	No. of frames with 3+ bit errors.
187 - 192	Floating Pt.	No. of frames with 0 bit errors.
193 - 198	Floating Pt.	No. of frames with 1 bit errors.
199 - 204	Floating Pt.	No. of frames with 2 bit errors.
205 - 210	Floating Pt.	No. of frames with 3+ bit errors.
211 - 216	Floating Pt.	No. of subcom slips in sail.
217 - 222	Floating Pt.	No. of subcom slips in wheel.
223 - 228	Floating Pt.	No. of subcom slips in digital.
229 - 234	Floating Pt.	No. of losses of frame sync.
235 - 240	Floating Pt.	No. of frames with fill.
241		Remainder of record blank.

ATTACHMENT IIDescription of Time Field

<u>Character</u>	<u>Representation</u>
1	Subcom Slip Flag*.
2 - 4	Not Used.
5 - 6	Day Count of Year (1 - 366).
7	Not Used.
8 - 12	Milliseconds of Day. (Time at word 1 of frame 1 of first subcom cycle in record).

\* The configuration of the subcom slip flag is as follows (bit 1 is least significant bit in character):  
1 = slip occurred, 0 = no slip.

<u>Bit</u>	<u>Representation</u>
1	Slip of sail subcom in subcom cycle 1.
2	Slip of wheel subcom in subcom cycle 1.
3	Slip of digital subcom in subcom cycle 1.
4	Slip of sail subcom in subcom cycle 2.
5	Slip of wheel subcom in subcom cycle 2.
6	Slip of digital subcom in subcom cycle 2.

ATTACHMENT IIIDescription of Flag Field

(Bit 1 is least significant bit in character)

Flag Field Character 1 (i.e. left-most character of field) contains:

<u>Bit</u>	<u>Representation</u>
1 - 4	No. of bit errors in main sync word.
5	1 = fill data in this frame, 0 = none.
6	1 = night, 0 = day.

Flag Field Character 2 (i.e., right-most character of field) contains:

<u>Bit</u>	<u>Representation</u>
1 - 6	No. of frame in subcom cycle (1 - 48).

ATTACHMENT IV

Description of Attitude Flag Field (Field is in 7094 format where bit 1 is LSB)

<u>Bit</u>	<u>Significance</u>
1	No $\phi, \eta, \beta_s$ data; data point was interpolated.
2	No $\omega$ data; data point was interpolated.
3	$\phi$ data out of bounds; data point was interpolated.
4	$\eta$ data out of bounds; data point was interpolated.
5	$\omega$ data out of bounds; data point was interpolated; $\beta_s$ is also interpolated by gyro pulse.
6	Pitch gas was activated.
7	No encoder pulse detected.
8	More than one encoder pulse detected.
9	Not used.
10	Not used.
11	Not used.
12	No pitch angle available.
13	No spin calculation possible.
14	No aspect calculation possible.

ATTACHMENT VDescription of Records With ORBIT Information

The first records in each file after the label record will contain the ORBIT information for the time interval covered by the data file. The first word in each record will contain all 1's (binary) indicating that ORBIT information will follow. The record will then be filled with 20-word data blocks containing the following information:

<u>Word No.</u>	<u>Representation</u>
1	Type of data item indicator: = 1 regular satellite data item. = 2 regular satellite data item. = 3 north point data item. = 4 descending node crossing. = 5 south point data item. = 6 sunlight entrance. = 7 sunlight exit.
2	Day count of year, time of data.
3	Seconds of day, time of data.
4	Satellite position vector x(km).
5	Satellite position vector y(km).
6	Satellite position vector z(km).
7	Satellite velocity vector x(km/sec).
8	Satellite velocity vector y(km/sec).
9	Satellite velocity vector z(km/sec).
10	Geodetic longitude (deg).
11	Geodetic latitude (deg).
12	Height above spheroid (km).
13	Solar vector $S_x$ (AU).
14	Solar vector $S_y$ (AU).
15	Solar vector $S_z$ (AU).
16	McIlwain L. Parameter L (earth radii).
17	Magnetic field strength B (gauss).
18	Right ascension of magnetic field vector (deg).
19	Declination of magnetic field vector (deg).
20	Ascending-node-crossing number.

All data with the exception of the geodetic information is given in Geocentric Equatorial Inertial (GEI) coordinates. Geodetic longitude is positive east of Greenwich, negative west of Greenwich. Geodetic northern latitudes are positive, and southern latitudes are negative. The geodetic information is based on the use of Fisher's Ellipsoid with an equatorial radius of 6378.166 km and a flattening of  $f = 1/298.25$ .



ATTACHMENT V

The 20-word data blocks will fill the first and succeeding records until the given ORBIT information covers the entire time interval of the data file. The 20-word data block will never be divided into parts to fit into the end of one data record and the beginning of another. The ending of ORBIT data is noted by filling the remainder of the data record with 88888888. (Floating point.) All ORBIT information is written in IBM-7094 floating point format.

## XXV. Description of Output

## A. Engineering Plot (Optional Output)

One hundredth of an inch represents 15.36 seconds or one major frame of data.

1. DAY-NIGHT : '-'=day ' '=night
2. ALTERNATE-NORMAL  
quadrant : '-'=alternate ' '=normal
3. MAGNETOMETER-GYRO  
reference : '-'=magnetometer ' '=gyro
4. CENTRAL DETECTOR  
ON-OFF : '-'=OFF ' '=ON
5. SHIELD ON-OFF : '-'=OFF ' '=ON
6. SCAN MODE : 'F'=full  
'I'=intermediate  
'S'=short

(The scan mode is printed at the start of each file and also after a change in scan mode.)

7. GAIN HIGH-LOW : 'H'=high 'L'=low

(Gain is printed at the start of each file and when gain changes.)

8. STEP NUMBER : The step number is calculated using the following constants and equations:

For gain = 1  
 $aV + b = \text{step}$

$$a\left(\frac{\text{cnts}}{51}\right) + b = \text{step}$$

$$1.5011 a + b = 0$$

$$3.565 a + b = 63$$

$$2.064 a = 63$$

$$a = 30.52 + .5 \text{ and}$$

$$b = -45.81 \text{ truncate}$$

For gain = 1/3

$$aV + b = \text{step}$$

$$a\left(\frac{\text{cnts}}{51}\right) + b = \text{step}$$

$$0.011 a + b = 0$$

$$2.074 a + b = 63$$

$$a = 30.53 \quad + .5 \text{ and}$$

$$b = -.336 \quad \text{truncate}$$

9. ELECTRONIC TEMPERATURE: Point should occur every 6 major frames.

$$\text{POINT} = \frac{\text{counts}}{255} * .25 \text{ in.}$$

10. DETECTOR TEMPERATURE: Same as No. 9.

11. CENTRAL DETECTOR: Same as No. 9.

12. SHIELD: Same as No. 9.

13. LOW VOLTAGE: Same as No. 9.

14. GAMMA COUNTS

.3 to 9 MeV SOLAR: Point should occur every major frame.

$$\text{POINT} = \frac{\text{counts } 0-120}{120} * 1.0 \text{ in.}$$

15. GAMMA COUNTS

.3 to 9 MeV BACKGROUND

: Same as No. 14.

16. GAMMA COUNTS > 9 MeV SOLAR

: Point should occur every major frame.

$$\text{POINT} = \frac{\text{counts } 0-60}{60} * 1.0 \text{ in.}$$

17. GAMMA COUNTS >

9 MeV BACKGROUND: Same as No. 16.

18. CUP SOLAR

: Point should occur every major frame.

$$\text{POINT} = \frac{\text{counts } 60-160}{100} * 1.0 \text{ in.}$$

19. CUP BACKGROUND: Same as No. 18.

20. SLAB SOLAR : Point should occur every major frame.  
POINT=  $\frac{\text{counts } 60-160}{100} * 1.0 \text{ in.}$
21. SLAB BACKGROUND : Same as No. 20.
22. DAY OF YEAR : Printed at start of each frame and whenever day changes.  
  
Time on the engineering plots is real time.
23. HOURS OF DAY : Hours 0-23 are printed -- day count changes at 0.
24. ORBIT NUMBER : This is the ascending node crossing number.

## B. Printed Output

### 1. Line Summary (Optional Output)

**Status Line**                      The status line is printed at the start of each file, after a long time gap, and whenever status, central detector on-off, or step number change. An asterisk is printed to the left of the field in which a change has occurred.

The status line consists of:

- |                     |       |   |
|---------------------|-------|---|
| a. DAY              | ----- | } This is the time of the major frame in which change occurs-- the time is the accumulation time.                                     |
| b. HOUR             |       |   |
| c. MINUTES          |       |   |
| d. SECONDS          | ----  |   |
| e. SCAN MODE        | :     | 'FULL', 'INTER', 'SHORT', or 'INVLD' (invalid).   |
| f. QUADRANT         | :     | 'A' - alternate 'N' - normal.   |
| g. DAY or NITE      | :     | Determined from status word.  |
| h. MAG or GYRO      | :     | Magnetometer or gyro reference (status word).   |
| i. SHLD/on-off      | :     | Determined from status word-- charged particle.   |
| j. CNTR/on-off      | :     | central detector -- determined from analog word (ASC 1-5).  |
| k. GAIN/hi-lo       | :     | Gain high or low -- determined from status word.  |
| l. STEP NUMBER      | :     | Analog word ASC 1-6.<br>The step number is calculated using same formulae and constants as in the plot section.                       |
| <b>Summary Line</b> | :     | One summary line is printed for 3 minutes of data; one full scan, 3 intermediate scans, or 6 short scans are represented by one line. |

- a. DAY -----
- b. HOUR
- c. MINUTES
- d. SECONDS-----

This is the first major frame in the section processed; it is the accumulation time.

- e. SOLAR

$$\text{DEAD TIME (seconds)} = \text{DEAD TIME} * .001515$$

$$\text{CLOCK TIME (seconds)} = \text{PHA CLOCK TIME} * \frac{1}{390.625} * \begin{matrix} 8 \text{ FULL} \\ 2 \text{ INTER} \\ 1 \text{ SHORT} \end{matrix}$$

$$\text{GAMMA SUM} = \sum \text{SOLAR GAMMA } .3 \text{ to } 9 \text{ MeV} / (\text{SOLAR CLOCK TIME} - \text{SOLAR DEAD TIME})$$

Following a radiation calibration:

$$\text{GAMMA SUM} = \sum \text{SOLAR GAMMA } .3 \text{ to } 9 \text{ MeV} / 100.0$$

$$\text{GAMMA LOG} = (\sum \text{SOLAR GAMMA } .3 \text{ to } 9 \text{ MeV} / \text{No. major frames}) * .1$$

$$\text{CUP RATE} = \text{Average solar cup rate} = \frac{\sum \text{SOLAR CUP}}{\text{No. major frames}}$$

$$\text{SLAB RATE} = \text{Average solar slab rate} = \frac{\sum \text{SOLAR SLAB}}{\text{No. major frames}}$$

- f. BACKGROUND

$$\text{DEAD TIME (seconds)} = \text{DEAD TIME} * .001515$$

$$\text{CLOCK TIME (seconds)} = \text{PHA CLOCK TIME} * \frac{1}{390.625} * \begin{matrix} 8 \text{ FULL} \\ 2 \text{ INTER} \\ 1 \text{ SHORT} \end{matrix}$$

$$\text{GAMMA SUM} = \sum \text{BACKGROUND GAMMA } .3 \text{ to } 9 \text{ MeV} / (\text{BACKGROUND-CLOCK TIME} - \text{BACKGROUND-DEAD TIME})$$

Following a radiation calibration

$$\text{GAMMA SUM} = 0.0$$

$$\text{GAMMA LOG} = (\sum \text{BACKGROUND GAMMA } .3 \text{ to } 9 \text{ MeV} / \text{No. major frames}) * .1$$

$$\text{CUP RATE} = \text{Average background cup rate} = \frac{\sum \text{BACKGROUND CUP}}{\text{No. major frames}}$$

$$\text{SLAB RATE} = \text{Average background slab rate} = \frac{\sum \text{BACKGROUND SLAB}}{\text{No. major frames}}$$

- g. AVERAGE DETECTOR  
TEMPERATURE
- h. AVERAGE ELECTRONIC  
TEMPERATURE
- i. AVERAGE CENTRAL  
HIGH VOLTAGE
- j. AVERAGE SHIELD  
HIGH VOLTAGE
- k. STATUS BITS:
  - SCAN MODE : 'F'=Full, 'I'=Intermediate,  
'S'=Short
  - QUADRANT : 'A'=Alternate, 'N'=Normal
  - ELECTRONIC  
CALIBRATION : 'E'=ON ' '=OFF
  - RADIATION  
CALIBRATION : 'R'=ON ' '=OFF
  - DAY-NIGHT  
indicator : 'D'=day 'N'=night
  - MAG-GYRO  
reference : 'M'=mag 'G'=gyro
  - CHARGED  
PARTICLE : 'P'=ON ' '=OFF
  - CENTRAL  
DETECTOR : 'C'=ON ' '=OFF
- l. AVERAGE LOW  
VOLTAGE
- m. NUMBER SYNC  
ERRORS IN  
SECTION PRO-  
CESSED
- n. ERROR MESSAGE  
NUMBER--if  
any

Cross-reference with error listing at end of line  
summary.

It is possible to have error messages interspersed with summary lines as well as having them printed in an error listing at the end of the run.

This would simply involve recompiling the main program with changes in the data statements containing the logical unit numbers for the different print sets used.

## 2. File, Tape, and Run Summaries (Output Each Run)

These summaries are an accumulation of statistics concerning the quantity and quality of the data for either a file, a tape, or a whole run.

Here is a list of the parameter accumulators that produce the necessary statistics:

- a. Count incrementer for sections processed.
- b. Count incrementer for sections bypassed.
- c. Count incrementer for complete (=12 major frames) sections processed.
- d. Count incrementer for complete sections processed with no analog flags.
- e. Count incrementer for complete sections processed with no MF flags.
- f. Number of scheduled status words missing or not recognized (processed sections).
- g. Number of time gaps (imbedded within a Decomm file) - total.
- h. Number of time gaps (imbedded within a Decomm file) - total "short".
- i. Major frames lost due to time gaps.
- j. Major frames bypassed.

For major frames processed:

- k. Total.
- l. No flagged analog.
- m. No flagged DSF (digital subcom).
- n. No flagged MF (main frame).



For X ray sequences selected for processing:

- o. Total selected.
- p. Total selected with no flags for digital subcom.
- q. Sync error instance count for frames processed.
- r. Sync error bit count for frames processed.
- s. Contradiction count.
- t. Day-night change count (of sections).
- u. Radioactive calibration count (of sections starting with R-Cal in progress).
- v. Electronic calibration count (of sections starting with E-Cal in progress).
- w. Total sections (processed or bypassed).

For file summary:

- x. I/O errors detected.
- y. Begin time parameters (yr, day of yr, hr, min, sec).
- z. End time parameters (yr, day of yr, hr, min, sec).
- aa. Tape sequence count.
- bb. File sequence count.

For tape summary:

- cc. I/O errors detected.
- dd. Earliest begin time parameters.
- ee. Latest end time parameters.
- ff. Tape sequence count.
- gg. Final file count.

For run summary:

- hh. I/O errors detected.

- ii. Earliest begin time parameters.
- jj. Latest end time parameters.
- kk. Final tape sequence count.
- ll. Total files summarized.
- mm. Files summarized but completely bypassed.

### 3. Data Anomaly Messages (Output Each Run)

Error messages are numbered so as to correspond to a summary line of the line summary. The number "nnnn" below represents this sequence count. It appears at the end of a summary line and at the beginning of the corresponding error message. There is also a severity code accompanying each message. This code '-s' simply tells the user the seriousness of the problem encountered. A severity code of 0 or 1 is quite common and reflects no serious problem. A severity code of 2 reflects greater problems such as time backslippage or I/O errors.

It is good to note that even if the line summary is not requested by the user, error messages and file, tape, and run summaries are provided by the program.

The messages that make up the error message listing are as follows:

- nnnn            = SEQUENCE COUNT (~~FOR PASS-I~~ RUN MADE AT  
nnmm HOURS OF yydd) OF FOLLOWING MESSAGE  
SERIES.
- s            I/O ERRORS OCCURRED ON n BLOCKS. IDENTIFIERS  
(TAPE, FILE, BLOCK, SENSE 1, SENSE 2) ARE  
SHOWN BELOW.  
t, f, b, x, z, --t<sub>n</sub> f<sub>n</sub> b<sub>n</sub> x<sub>n</sub> z<sub>n</sub>
- s            PROGRAM TERMINATION REQUESTED BY FEEDER.  
CALL ARGUMENTS = (aa, bb, cc, dd, ee, ff,  
gg), IHEXPN = hh, QHDERR = p, QHTAPN = q.
- s            TIME GAP (EXCEEDING nnn MAJOR FRAMES) HAS  
OCCURRED IN DATA ACCEPTED BY PASS-I PROGRAM.
- s            TIME BACKSLIPPAGE IN TAPE INPUT DATA. PRE-  
VIOUS IDENTIFIERS = (DAY = ddd, SECONDS =  
sssss.ss, TIME = tt, FILE = ff, BLOCK = bbb).  
CURRENT IDENTIFIERS = (DAY = ddd, SECONDS =  
sssss.ss, TAPE = tt, FILE = ff, BLOCK = bbb).
- s            PROGRAM TERMINATED BY EXCESS FILES. PANE  
15 IDENTIFIERS = (TAPE = tt, FILE = ff,  
BLOCK = bbb).

-S DATA ACCEPTANCE SUSPENDED DUE TO HIGH VOLTAGE BEING OFF.

-S STATUS WORD DISCONTINUITY. SUBROUTINE PUZZLE CALLED TO ATTEMPT SCAN MODE IDENTIFICATION.

-S ILLEGAL GAP IN STATUS WORDS. IDENTIFIERS  
= (IB = bb, IA = aa, TAPE = tt, FILE = ff,  
BLK = bbb, DAY = ddd, SECONDS = ssssss.ss).  
    111          1  
    mmm          m  
    nnn          n

-S SUBROUTINE PUZZLE RESORTING TO FREE-WHEELING FOR FINAL SCAN MODE ID ATTEMPT.

-S COMPLETE BREAKDOWN IN SCAN MODE ID. IDENTIFIERS  
= (TAPE = tt, FILE = ff, BLK = bb, DAY = ddd,  
SECONDS = ssssss.ss).

### C. Calibration Cards (Optional Output)

[illegible]

YEAR

MONTH

DAY

HOUR

MINUTES

SECONDS

Midpoint of Accumulation Time

CHANNEL (1)

Channel number of 1st peak channel.

CHANNEL (2)

Channel number of 2nd peak channel.

CHANNEL (3)

Channel number of 3rd peak channel.

COUNTS (1)

Sum of counts around 1st peak  
channel.

COUNTS (2)

Sum of counts around 2nd peak  
channel.

COUNTS (3)

Sum of counts around 3rd peak  
channel.

ID

= '200' for calibration cards.

#### D. X-Ray Data Tape (Optional Output)

This is a 9-track tape containing interpreted x-ray values. For missing major frames, a filler of -1's is used. The time parameter reflects real time.

The logical record length is 76 characters and the block size is 1520 characters.

Here is a list of the parameters on the x-ray tape. (All parameters are 16-bit integers except for numbers 7,8,25, 26,27, and 28 which are 32-bit floating point words):

1. YEAR of 1st major frame.
2. YEAR of 2nd major frame.
3. DAY OF YEAR of 1st major frame.
4. DAY OF YEAR of 2nd major frame.
5. HOUR OF DATA of 1st major frame.
6. HOUR OF DATA of 2nd major frame.
7. MINUTES OF DATA of 1st major frame.
8. MINUTES OF DATA of 2nd major frame.
9. SECONDS OF DATA of 1st major frame.
10. SECONDS OF DATA of 2nd major frame.
11. STATUS WORD of 12 bits in half-word of 1st major frame.
12. SOLAR 7.5 - 15 keV
13. SOLAR 15 - 30 keV
14. SOLAR 30 - 60 keV
15. SOLAR 60 - 120 keV
16. BACKGROUND 7.5 - 15 keV
17. BACKGROUND 15 - 30 keV
18. BACKGROUND 30 - 60 keV
19. BACKGROUND 60 - 120 keV

20. SOLAR QUADRANT TIME of 1st major frame.
21. SOLAR QUADRANT TIME of 2nd major frame.
22. BACKGROUND QUADRANT TIME of 1st major frame.
23. BACKGROUND QUADRANT TIME of 2nd major frame.
24. STATUS WORD of 12 bits in half word of 2nd major frame.
25. SPIN RATE of 1st major frame.
26. SPIN RATE of 2nd major frame.
27. AZIMUTH ENCODER TIME of 1st major frame.
28. AZIMUTH ENCODER TIME of 2nd major frame.
29. ELECTRONIC CALIBRATION 1 - on, 2 - off, 0 - no major frame.
30. RADIATION CALIBRATION 1 - on, 2 - off, 0 - no major frame.
31. DAY - NIGHT INDICATOR 1 - day, 2 - night, 0 - no status.
32. EXTRA WORD.



YEAR -----  
MONTH  
DAY  
HOURS  
MINUTES  
SECONDS -----

} This is the accumulation time of  
the last scan to be processed.

Bo/BK

This field indicates whether to  
stop processing at the background  
scan or after processing both  
background and solar.

ID

= '100' for good time cards.



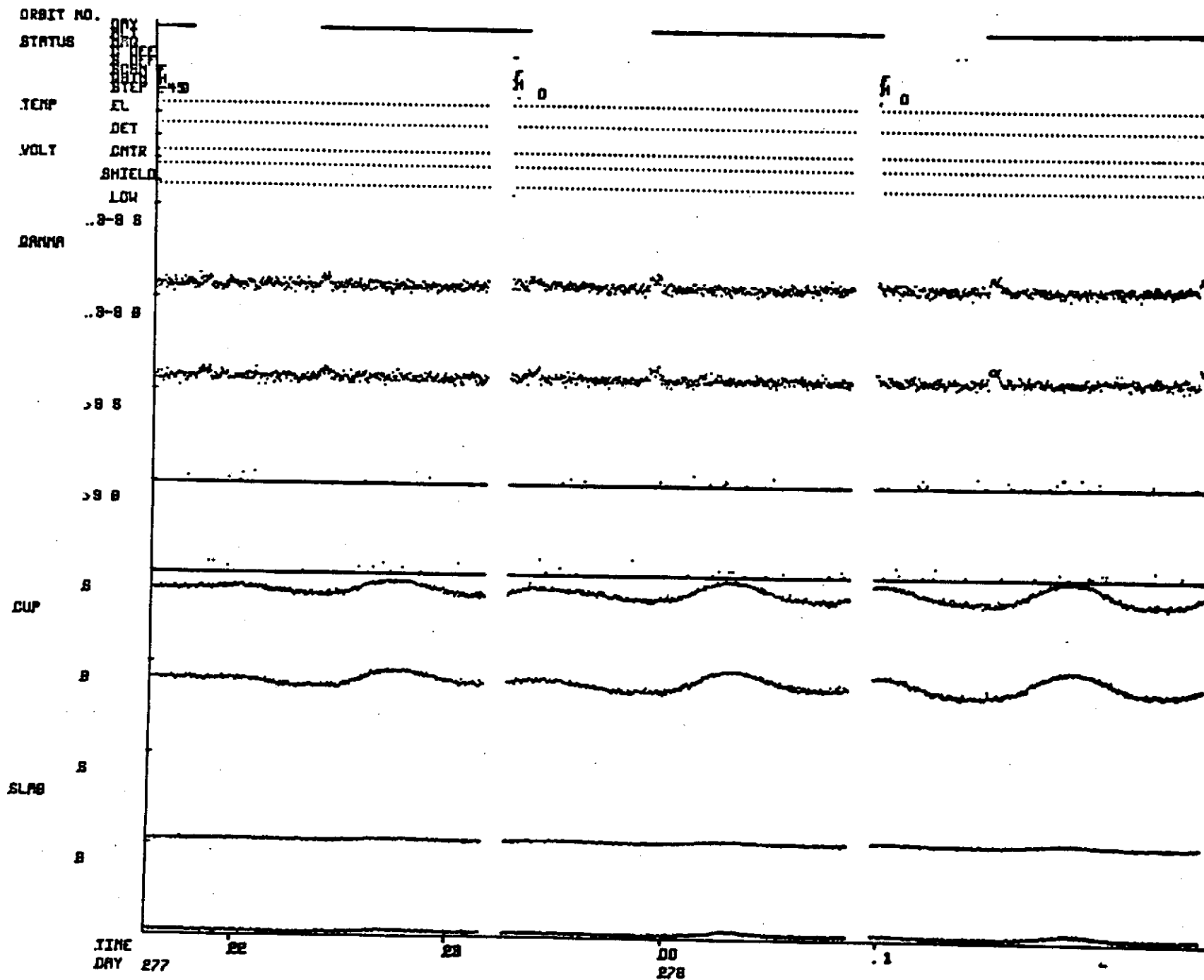
## F. Main Frame Data Tape (Optional Output)

All parameters on this tape are 16-bit integers. The logical word length is 908 characters with a block size of 908 characters.

<p>HYR : Year  HMO : Month  HDA : Day  HJUL : Julian Date  HHR : Hour  HMN : Minutes  HSE : Seconds</p>	<p>: This time is the read-out time of the first major frame processed in this section. The time is the read-out time of the main frame data and the real time of analog words and main frame status. When the first major frame is missing, the read-out time for the quadrant scan is calculated if possible.</p>
<p>HAN(60) : Analog words</p>	<p>: Optimally, there are 6 major frames processed per quadrant scan and this field contains the 60 analog words for the scan, 10 words from each major frame processed. When there are major frames missing, -1's are substituted in the area where the gap occurred, if possible. (This is not usually possible at the beginning of a file.)</p>
<p>HMFDAT(384) : Main Frame Date</p>	<p>: This file contains the main frame data corresponding to a quadrant scan of 6 major frames. If there is a time gap, -1's fill the area where the gap occurred if this is possible. The main frame data was accumulated 184.32 seconds earlier than the above given read-out time for a background quadrant and 276.48 seconds earlier for a solar quadrant.</p>
<p>HTGAP : Time gap indicator</p>	<p>: This is an indicator that is set to '0' when 6 major frames are being processed and set to '1' when there is data missing in the section.</p>

HFCNT : Frame counter : This is the major frame counter.  
It indicates how many major  
frames are being processed in  
the section.

HEXTRA : Extra field : This indicator is set to zero  
and is an extra field added to  
the tape for possible future  
use.



Engineering Plot

### Line Summary

RJN DATE	YR/DAY OF YR	73355	HRS/MIN 2340	*FULL	*NJQAD	*NITE	*GYRO	*SHLD/ON	CNTR/	*GAIN/HI	STEP/	O							
TIME	TD	TC	GAMMA	CUP	SLAB	TO	TC	GAMMA	CUP	SLAB	TEMP	TEMP	CENT	CP	STATUS	LVPS	TG	SYNC	MESSAGE
	SOLAR		SUM LOG			BACKGROUND		SUM LOG			DET	EL	HI	V HI	V	BITS			ERRS
216 21 33 4.1	FULL	NJQAD	NITE	GYRO	SHLD/ON	CNTR/ON	GAIN/HI	STEP/	O										
216 21 33 19.4	FULL	NJQAD	NITE	GYRO	SHLD/ON	CNTR/ON	GAIN/HI	STEP/	63										
216 21 32 33.3 F 0.0 45.6	8.1	3.7	157.6	68.5	0.0	0.0	0.0	3.3	156.0	67.2	14.4	13.9	172	204	N	NGPC	234	1	0
216 21 35 6.9 F 0.9 45.6	9.9	3.7	157.8	68.2	23.6	45.6	18.1	3.3	156.1	67.1	14.4	13.8	172	204	N	NGPC	235	0	0
216 21 38 11.3 F 1.2 46.1	9.8	3.6	156.8	68.0	23.4	45.6	19.3	3.6	155.3	67.0	14.4	13.9	172	204	N	NGPC	234	0	0
216 21 41 15.6 F 2.3 45.6	20.8	3.7	154.5	67.5	22.4	45.6	18.1	3.5	152.5	66.4	14.4	13.9	172	204	N	NGPC	234	0	0
216 21 44 19.9 F 2.4 45.6	17.9	3.6	149.8	66.5	19.2	45.6	16.6	3.6	147.6	55.2	14.4	13.3	172	204	N	NGPC	235	0	0
216 21 48 56.4	FULL	NJQAD	*DAY	GYFO	SHLD/ON	CNTR/ON	GAIN/HI	STEP/	63										
216 21 47 24.2 F 16.4 45.6	15.1	3.7	144.9	65.7	14.9	45.6	14.1	3.5	142.3	55.7	14.4	13.8	172	204	N	NGPC	234	0	0
216 21 50 28.5 F 12.8 44.6	16.8	4.5	141.3	65.1	11.2	44.6	16.1	4.5	139.5	65.3	14.4	13.9	172	204	N	DGPC	234	0	0
216 21 53 32.9 F 10.7 7.4	4.3	3.6	139.0	64.9	0.0	0.0	0.0	3.6	136.8	64.8	14.4	13.8	172	204	N	DGPC	235	0	0
216 21 56 37.2 F 9.6 45.6	12.3	3.7	137.8	64.8	9.1	45.6	12.2	3.7	137.1	64.8	14.4	13.9	172	204	N	DGPC	235	0	0
216 21 59 41.5 F 9.2 45.6	12.1	3.7	139.5	65.0	9.1	45.6	12.0	3.6	136.8	64.9	14.4	13.9	172	204	N	DGPC	235	0	0
216 22 2 45.8 F 9.9 45.6	11.7	3.5	133.3	65.2	9.1	45.6	12.0	3.7	137.4	64.8	14.4	13.9	172	204	N	DGPC	235	0	0
216 22 5 50.1 F 9.5 46.1	12.0	3.6	139.5	65.2	9.2	45.6	12.1	3.7	139.3	55.0	14.4	13.9	172	204	N	DGPC	235	0	0
216 22 8 54.5 F 10.4 45.6	12.0	3.5	141.3	65.3	10.3	45.6	12.2	3.6	142.0	55.2	14.4	13.9	172	204	N	DGPC	234	0	0
216 22 11 58.8 F 10.9 45.6	12.4	3.6	143.5	65.5	11.6	45.6	13.0	3.7	143.6	55.8	14.4	13.9	172	204	N	DGPC	234	0	0
216 22 15 3.1 F 12.6 46.1	12.8	3.6	147.0	65.1	13.6	45.6	13.9	3.7	143.4	55.2	14.4	13.9	172	204	N	DGPC	235	0	0
216 22 18 7.4 F 15.0 45.6	13.7	3.5	149.4	65.2	15.2	45.6	14.8	3.6	151.9	66.3	14.4	13.9	172	204	N	DGPC	234	0	0
216 22 21 11.7 F 16.7 45.6	14.0	3.4	152.3	66.7	19.3	46.1	16.9	3.8	153.8	67.3	14.4	13.8	172	204	N	DGPC	234	0	0
216 22 24 15.1 F 18.8 45.6	15.5	3.5	151.5	65.6	20.8	45.0	17.8	3.7	1										

[illegible]

# File, Tape, and Run Summary

FILE SUMMARY FOR 1TH FILE READ FROM TAPE=(COUNT= 1, VOLSER=EXD224) DURING PASS-I RUN STARTED AT 2340 HOURS OF 73355.  
DATA TIMES FROM 1972Y 216D 21H 35M 37S TO 1972Y 216D 23H 7M 47S. 0 I/O ERRORS. 0 SC SYNC BITS IN ERROR IN 0 MINOR FRAMES.

* SECTION	QUANTITY	* MAJOR-FRAME	QUANTITY	* SECTION QUALITY	* MAJOR-FRAME QUALITY	* XRAY DATA
* PRO- BY-	*	PRO- BY-	T-GAP	* PRO- COM- CLEAN	CLEAN	* PRO- CLEAN CLEAN CLEAN
* TOTAL CESS	PASS	* TOTAL CESS	PASS LOSS	* CESS PLETE WHEEL	MF	* CESS WHEEL DSF MF
CNT	* 30	30	0	* 358	358	0
HRS	* 1.53	1.53	1.53	0.0	0.0	0.0
PCT	* 93.4	93.4	93.4	0.0	0.0	0.0

FILE SUMMARY FOR 2TH FILE READ FROM TAPE=(COUNT= 1, VOLSER=EXD224) DURING PASS-I RUN STARTED AT 2340 HOURS OF 73355.  
DATA TIMES FROM 1972Y 216D 23H 13M 55S TO 1972Y 217D 0H 46M 55S. 0 I/O ERRORS. 1 SC SYNC BITS IN ERROR IN 1 MINOR FRAMES.

* SECTION	QUANTITY	* MAJOR-FRAME	QUANTITY	* SECTION QUALITY	* MAJOR-FRAME QUALITY	* XRAY DATA
* PRO- BY-	*	PRO- BY-	T-GAP	* PRO- COM- CLEAN	CLEAN	* PRO- CLEAN CLEAN CLEAN
* TOTAL CESS	PASS	* TOTAL CESS	PASS LOSS	* CESS PLETE WHEEL	MF	* CESS WHEEL DSF MF
CNT	* 30	30	0	* 358	358	0
HRS	* 1.53	1.53	1.51	0.0	0.02	0.0
PCT	* 93.4	93.4	92.3	0.0	1.0	0.0

FILE SUMMARY FOR 3TH FILE READ FROM TAPE=(COUNT= 1, VOLSER=EXD224) DURING PASS-I RUN STARTED AT 2340 HOURS OF 73355.  
DATA TIMES FROM 1972Y 217D 0H 52M 13S TO 1972Y 217D 2H 1M 55S. 0 I/O ERRORS. 303 SC SYNC BITS IN ERROR IN 72 MINOR FRAMES.

* SECTION	QUANTITY	* MAJOR-FRAME	QUANTITY	* SECTION QUALITY	* MAJOR-FRAME QUALITY	* XRAY DATA
* PRO- BY-	*	PRO- BY-	T-GAP	* PRO- COM- CLEAN	CLEAN	* PRO- CLEAN CLEAN CLEAN
* TOTAL CESS	PASS	* TOTAL CESS	PASS LOSS	* CESS PLETE WHEEL	MF	* CESS WHEEL DSF MF
CNT	* 23	23	0	* 269	268	0
HRS	* 1.15	1.15	1.14	0.0	0.00	0.0
PCT	* 92.0	92.0	91.5	0.0	0.3	0.0

FILE SUMMARY FOR 4TH FILE READ FROM TAPE=(COUNT= 1, VOLSER=EXD224) DURING PASS-I RUN STARTED AT 2340 HOURS OF 73355.  
DATA TIMES FROM 1972Y 217D 2H 7M 28S TO 1972Y 217D 3H 39M 7S. 0 I/O ERRORS. 0 SC SYNC BITS IN ERROR IN 0 MINOR FRAMES.

* SECTION	QUANTITY	* MAJOR-FRAME	QUANTITY	* SECTION QUALITY	* MAJOR-FRAME QUALITY	* XRAY DATA
* PRO- BY-	*	PRO- BY-	T-GAP	* PRO- COM- CLEAN	CLEAN	* PRO- CLEAN CLEAN CLEAN
* TOTAL CESS	PASS	* TOTAL CESS	PASS LOSS	* CESS PLETE WHEEL	MF	* CESS WHEEL DSF MF
CNT	* 31	31	0	* 356	356	0
HRS	* 1.52	1.52	1.52	0.0	0.0	0.0
PCT	* 93.3	93.3	93.3	0.0	0.0	0.0

FILE SUMMARY FOR 5TH FILE READ FROM TAPE=(COUNT= 1, VOLSER=EXD224) DURING PASS-I RUN STARTED AT 2340 HOURS OF 73355.  
DATA TIMES FROM 1972Y 217D 3H 45M 15S TO 1972Y 217D 5H 16M 54S. 0 I/O ERRORS. 0 SC SYNC BITS IN ERROR IN 0 MINOR FRAMES.

* SECTION	QUANTITY	* MAJOR-FRAME	QUANTITY	* SECTION QUALITY	* MAJOR-FRAME QUALITY	* XRAY DATA
* PRO- BY-	*	PRO- BY-	T-GAP	* PRO- COM- CLEAN	CLEAN	* PRO- CLEAN CLEAN CLEAN
* TOTAL CESS	PASS	* TOTAL CESS	PASS LOSS	* CESS PLETE WHEEL	MF	* CESS WHEEL DSF MF
CNT	* 31	31	0	* 358	358	0
HRS	* 1.53	1.53	1.53	0.0	0.0	0.0
PCT	* 93.9	93.9	93.9	0.0	0.0	0.0

FILE SUMMARY FOR 6TH FILE READ FROM TAPE=(COUNT= 1, VOLSER=EXD224) DURING PASS-I RUN STARTED AT 2340 HOURS OF 73355.  
DATA TIMES FROM 1972Y 217D 5H 23M 39S TO 1972Y 217D 6H 54M 47S. 0 I/O ERRORS. 0 SC SYNC BITS IN ERROR IN 0 MINOR FRAMES.

* SECTION	QUANTITY	* MAJOR-FRAME	QUANTITY	* SECTION QUALITY	* MAJOR-FRAME QUALITY	* XRAY DATA
* PRO- BY-	*	PRO- BY-	T-GAP	* PRO- COM- CLEAN	CLEAN	* PRO- CLEAN CLEAN CLEAN
* TOTAL CESS	PASS	* TOTAL CESS	PASS LOSS	* CESS PLETE WHEEL	MF	* CESS WHEEL DSF MF
CNT	* 31	31	0	* 356	356	0
HRS	* 1.52	1.52	1.52	0.0	0.0	0.0
PCT	* 93.8	93.8	93.8	0.0	0.0	0.0

FILE SUMMARY FOR 7TH FILE READ FROM TAPE=(COUNT= 1, VOLSER=EXD224) DURING PASS-I RUN STARTED AT 2340 HOURS OF 73355.  
DATA TIMES FROM 1972Y 217D 7H 1M 55S TO 1972Y 217D 8H 33M 14S. 0 I/O ERRORS. 3 SC SYNC BITS IN ERROR IN 1 MINOR FRAMES.

* SECTION	QUANTITY	* MAJOR-FRAME	QUANTITY	* SECTION QUALITY	* MAJOR-FRAME QUALITY	* XRAY DATA
* PRO- BY-	*	PRO- BY-	T-GAP	* PRO- COM- CLEAN	CLEAN	* PRO- CLEAN CLEAN CLEAN
* TOTAL CESS	PASS	* TOTAL CESS	PASS LOSS	* CESS PLETE WHEEL	MF	* CESS WHEEL DSF MF

# File, Tape, and Run Summary

DATA TIMES FROM 1972Y 219D 2H 4M 26S TO 1972Y 219D 3H 36M 4S. 0 I/O ERRORS. 0 SC SYNC BITS IN ERROR IN 0 MINOR FRAMES.

* SECTION QUANTITY *				MAJOR-FRAME QUANTITY *				SECTION QUALITY *				MAJOR-FRAME QUALITY *				XRAY DATA	
* PRO- BY- *				* PRO- BY- T-GAP *				* PRO- COM- CLEAN CLEAN *				* PRO- CLEAN CLEAN CLEAN *				* SE- COMPG	
* TOTAL CESS PASS *				* TOTAL CESS PASS LOSS *				* CESS PLETE WHEEL MF *				* CESS WHEEL DSF MF *				* LECT CLEAN	
CNT *	30	30	0	350	350	0	0	30	29	28	28	356	355	355	355	178	177
HRS *	1.52			1.52	1.52	0.0	0.0	1.43	1.43	1.43		1.52	1.51	1.51	1.51	1.52	1.51
PCT *	93.3			93.3	93.3	0.0	0.0	88.1	88.1	88.1		93.3	93.1	93.1	93.1	93.3	92.8

TAPE SUMMARY FOR 34 FILES READ FROM TAPE=ICOUNT= 1, VOLSER=EXD224) DURING PASS-I RUN STARTED AT 1700 HOURS OF 74014.

DATA TIMES FROM 1972Y 216D 21H 35M 37S TO 1972Y 219D 3H 36M 4S. 0 I/O ERRORS. 5117 SC SYNC BITS IN ERROR IN 716 MINOR FRAMES.

* SECTION QUANTITY *				MAJOR-FRAME QUANTITY *				SECTION QUALITY *				MAJOR-FRAME QUALITY *				XRAY DATA	
* PRO- BY- *				* PRO- BY- T-GAP *				* PRO- COM- CLEAN CLEAN *				* PRO- CLEAN CLEAN CLEAN *				* SE- COMPG	
* TOTAL CESS PASS *				* TOTAL CESS PASS LOSS *				* CESS PLETE WHEEL MF *				* CESS WHEEL DSF MF *				* LECT CLEAN	
CNT *	988	988	0	11461	11455	0	6	988	922	888	883	11455	11295	11249	11177	5719	5547
HRS *	48.90			48.90	48.87	0.0	0.03	47.21	45.47	45.21		48.87	48.19	48.00	47.69	48.80	47.33
PCT *	90.4			90.4	90.3	0.0	0.0	87.2	84.0	83.6		90.3	89.1	88.7	88.1	90.2	87.5

RUN SUMMARY FOR 34 FILES READ FROM 1 TAPES DURING PASS-I RUN STARTED AT 1700 HOURS OF 74014.

DATA TIMES FROM 1972Y 216D 21H 35M 37S TO 1972Y 219D 3H 36M 4S. 0 I/O ERRORS. 5117 SC SYNC BITS IN ERROR IN 716 MINOR FRAMES.

* SECTION QUANTITY *				MAJOR-FRAME QUANTITY *				SECTION QUALITY *				MAJOR-FRAME QUALITY *				XRAY DATA	
* PRO- BY- *				* PRO- BY- T-GAP *				* PRO- COM- CLEAN CLEAN *				* PRO- CLEAN CLEAN CLEAN *				* SE- COMPG	
* TOTAL CESS PASS *				* TOTAL CESS PASS LOSS *				* CESS PLETE WHEEL MF *				* CESS WHEEL DSF MF *				* LECT CLEAN	
CNT *	988	988	0	11461	11455	0	6	988	922	888	883	11455	11295	11249	11177	5719	5547
HRS *	48.90			48.90	48.87	0.0	0.03	47.21	45.47	45.21		48.87	48.19	48.00	47.69	48.80	47.33
PCT *	90.4			90.4	90.3	0.0	0.0	87.2	84.0	83.6		90.3	89.1	88.7	88.1	90.2	87.5

## PASS-1 GOOD TIME CARDS

72	8	3	21	36:39.1	BO	72	8	3	23	2:40.1	BO
72	8	3	23	11:52.2	SO	72	8	3	23	11:52.2	BO
72	8	3	23	14:56.9	SO	72	8	4	0	40:57.8	BO
72	8	4	0	50:10.4	SO	72	8	4	1	54:41.2	BO
72	8	4	2	6:58.0	BO	72	8	4	3	32:58.9	BO
72	8	4	3	45:15.7	BO	72	8	4	5	11:16.7	BO
72	8	4	5	20:50.5	SO	72	8	4	6	49:55.8	BO
72	8	4	6	58:47.0	SO	72	8	4	8	27:52.2	BO
72	8	4	8	37: 4.7	SO	72	8	4	10	9:14.3	BK
72	8	4	10	18:26.8	BO	72	8	4	11	47:32.1	BK
72	8	4	11	56:44.6	BO	72	8	4	13	28:54.2	BK
72	8	4	13	38: 6.7	SO	72	8	4	15	7:12.0	BK
72	8	4	15	16:24.5	BO	72	8	4	16	45:29.7	BK
72	8	4	16	54:42.2	BO	72	8	4	18	23:47.5	BK
72	8	4	18	33: 0.3	BO	72	8	4	18	54:30.5	BO
72	8	4	19	6:47.4	BO	72	8	4	20	32:48.3	BO
72	8	4	20	42: 0.8	SO	72	8	4	22	11: 6.1	BO
72	8	4	22	23:22.9	BO	72	8	4	23	49:23.9	BO
72	8	5	0	1:40.7	BO	72	8	5	1	27:41.6	BO
72	8	5	1	36:54.2	SO	72	8	5	2	16:50.4	BO
72	8	5	2	22:59.0	BO	72	8	5	2	44:29.3	BK
72	8	5	2	53:41.8	BO	72	8	5	4	22:47.1	BK
72	8	5	4	31:59.6	BO	72	8	5	6	1: 4.8	BK
72	8	5	6	10:17.3	BO	72	8	5	7	36:18.3	BO
72	8	5	9	14:35.6	SO	72	8	5	10	59: 2.5	BK
72	8	5	11	8:15.1	BO	72	8	5	12	0:28.6	BO
72	8	5	12	9:41.1	SO	72	8	5	13	38:46.3	BK
72	8	5	13	47:58.8	BO	72	8	5	15	17: 4.1	BK
72	8	5	15	26:16.7	BO	72	8	5	16	30:47.5	BK
72	8	5	16	39:60.0	BO	72	8	5	18	6: 0.9	BK
72	8	5	18	15:13.4	BO	72	8	5	19	41:14.4	BO
72	8	5	19	50:26.9	SO	72	8	5	21	22:36.5	BK
72	8	5	21	31:49.0	BO	72	8	5	22	57:49.9	BO
72	8	5	23	10: 6.7	BO	72	8	6	0	36: 7.7	BO
72	8	6	0	45:20.3	SO	72	8	6	1	52:55.3	BO
72	8	6	2	2: 7.8	SO	72	8	6	3	31:13.1	BK



## Section 3

### CAL1

	<u>Page</u>
I. Introduction	3-1
II. Macro Systems Flowchart	3-2
III. Job Stream Schematic	3-3
IV. Job Control Language	3-4
V. CAL1 (Main Program)	3-5
VI. Subroutine FORTIM (IRUNDA,IRUNT)	3-9
VII. Subroutine FPUT (IDSN,ICMD,IRSLT,IOVEC)	3-10
VIII. Description of Input	3-12
IX. Description of Output	3-13
X. Sample Output Listing	

## I. Introduction

This program reads a series of input cards, the first of which contains the  $n_0$  to be used in the calculation of the gain constant (for our purposes, a constant  $n_0$  of 80.2 was used). The other input cards are calibration cards generated by Pass-I. Each card contains midpoint of accumulation time, three peak channels, and three peak counts. From this information CAL1 calculates an  $n_0$  for each input card from the relation:

$$n_0 = \frac{\sqrt{E_1} n_2 - \sqrt{E_2} n_1}{\sqrt{E_2} - \sqrt{E_1}}$$

An average  $n_0$  is found for all the cards read.

The program then calculates a gain constant for each card using weighted average:

$$C = \sum_i w_i \frac{E_i}{(n_i + n_0)^2}$$

where:

$$E_1 = 1.17323 \text{ MeV}$$

$$E_2 = 1.33248 \text{ MeV}$$

$$E_3 = 2.525 \text{ MeV}$$

$$w_i = \frac{N_i}{N_1 + N_2 + N_3}$$

$N_i$  = number of counts under the peak at energy  $E_i$

$n_i$  = the channel that corresponds to the energy  $E_i$ .

An error statistic is calculated for each calibration in the following way:

For  $i = 1, 3$

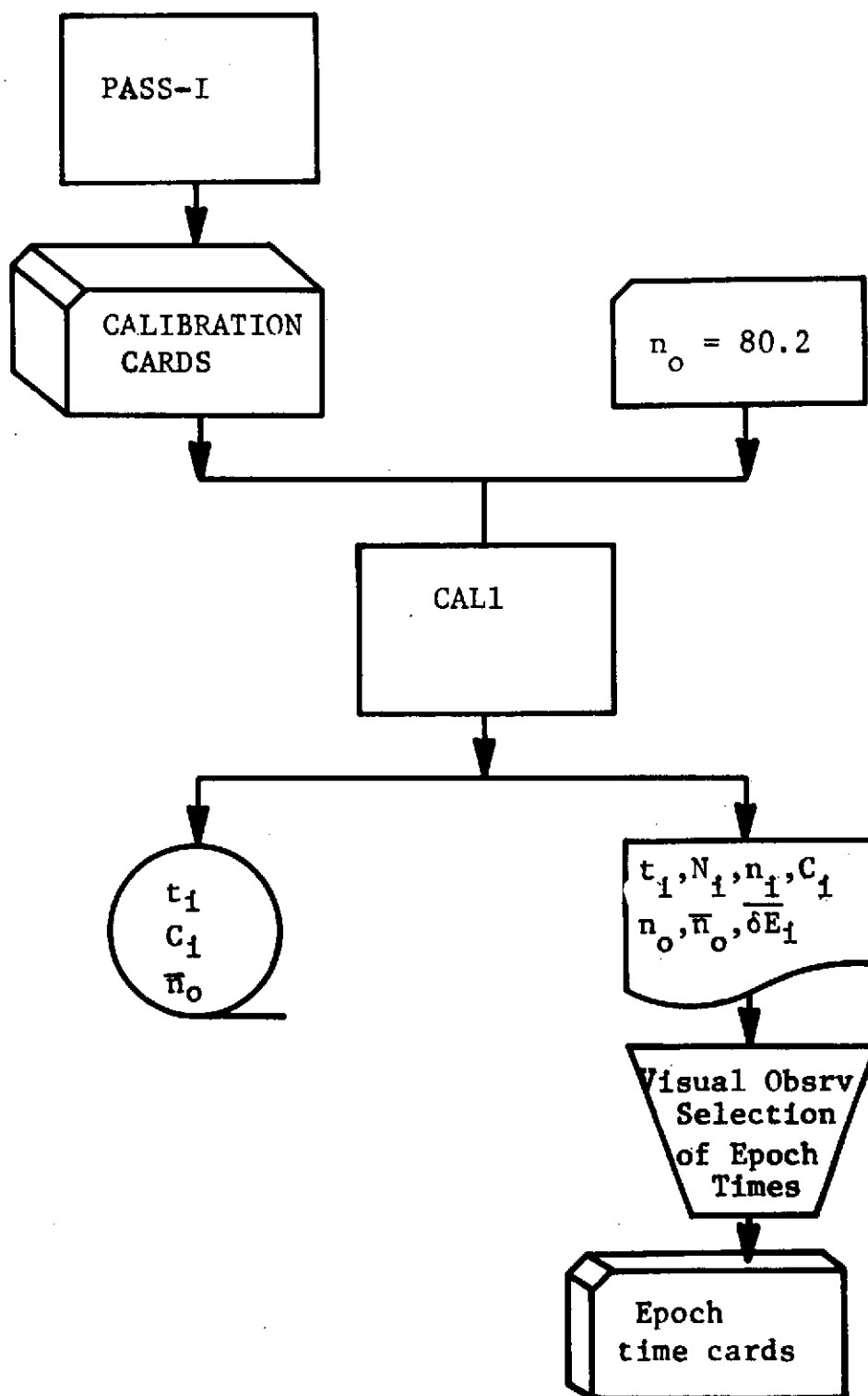
$$ER_i = (E_i - C(n_i + n_0)^2) / E_i$$

$$\text{ERROR} = \sqrt{\sum_{i=1,3} (ER_i)^2}$$

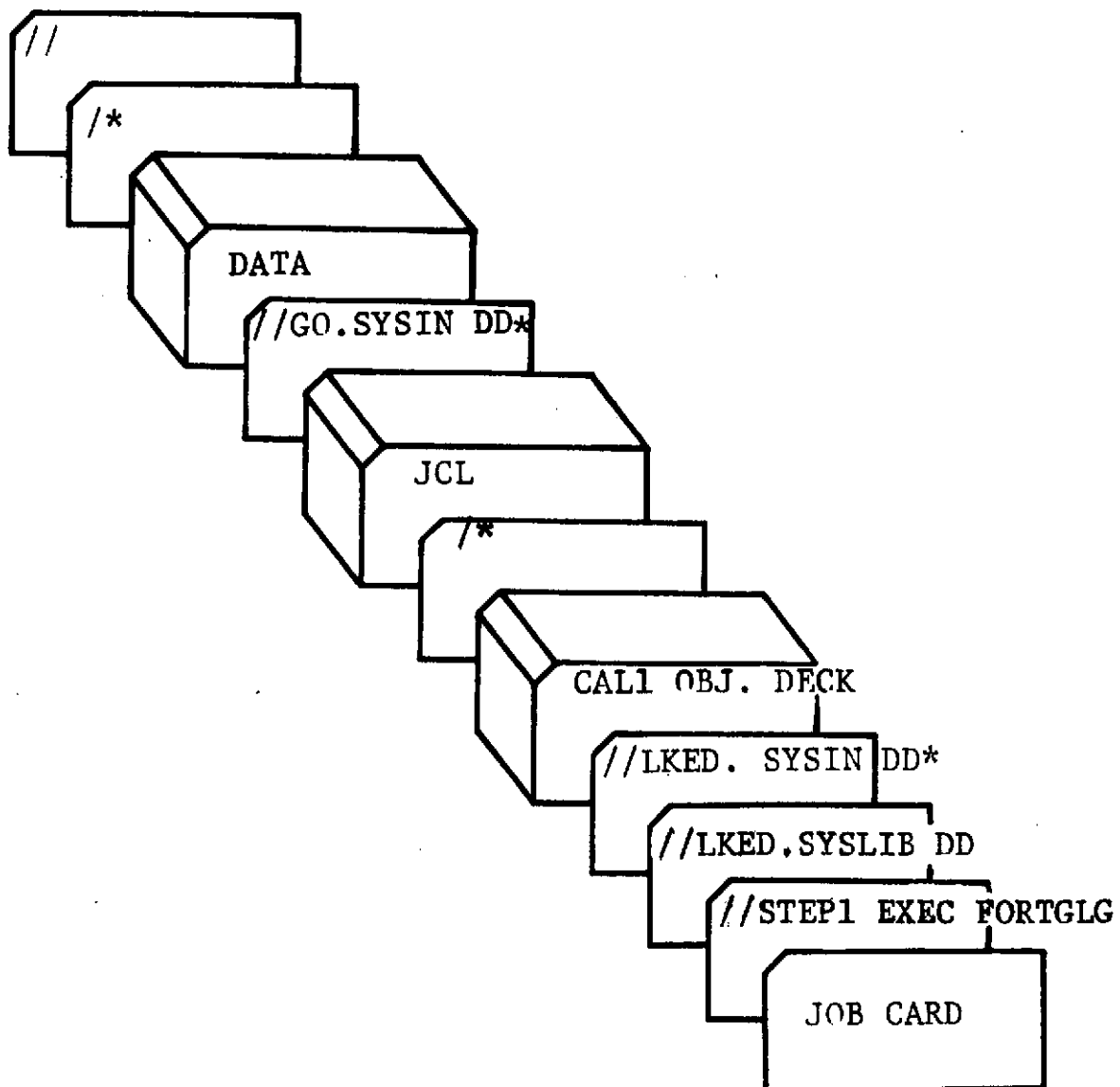
A magnetic tape record is generated for each calibration card and contains time at midpoint of accumulation,  $C$ , and  $n_0$ . The program then prints a listing of all the cards read - time,  $C$ , average  $n_0$ ,  $I$ ,  $N_0$ , channels, counts, and the error indicator.

This listing is reviewed by our project engineer and epoch times are derived from this source in conjunction with the output from the Van Allen Predict program (see Section 6).

## II. Macro Systems Flowchart (CAL1)



## III. Job Stream Schematic (CAL1)



## IV. JOB CONTROL LANGUAGE - CAL1

```
//          T548-12), 'S.CROTEAU,CAL1',CLASS=F,TIME=03,REGION=100K
// *LINECNT 62
//STEP1 EXEC  FORTGLG
//LKED.SYSLIB DD DSN=USER.T317.SUBLIB,DISP=SHR
//          DD DSN=USER.T548.SUBLIB,DISP=SHR
//          DD DSN=SYS1.FORTLIB,DISP=SHR
//          DD DSN=USER.SSPLIB,DISP=SHR
//LKED.SYSIN DD *
```

\*\*\* CAL1 OBJECT DECK \*\*\*

```
/*
//GO.FPUT01 DD UNIT=2400,DISP=(NEW,KEEP),LABEL=(1,SL),
//  DSN=CAL1,DCB=(RECFM=FB,LRECL=36,BLKSIZE=1800),
//          VOLUME=SER=CC0922
//GO.SYSUDUMP DD SYSOUT=A
//GO.SYSIN DD *
80.2
/*
//
```

## V. CAL1 (Main Program)

Program Language : Fortran IV, G-Level  
Date Written : May 1972  
Programmer : S. Croteau  
Installation : University of New Hampshire  
Computer : IBM-360-H50  
Subprograms Called : FPUT, FORTIM

The purpose of this program is to calculate  $c$  for each calibration using a constant  $n_0$  of 80.2. An error statistic is also found which is the error in the least squares fit to the three energy points as a function of channel number. The reasons for calculating this error statistic are:

1. To make sure the computer hasn't picked up a spurious peak (e.g. bit error).
2. To ascertain that the instrument has remained linear.

The program first reads in the calibration cards generated by PASS-I. It then calculates:

1. An  $n_0$  for each card.
2. An average  $n_0$  for all the cards.
3.  $C$  using an  $n_0$  of 80.2.
4. The error statistic.

The results of these calculations are stored in tables. A tape record is then written for each calibration and a report is printed of all input and calculated output values.

## A. Unlabelled Common

JYR : Integer full word variable containing the year at midpoint of accumulation.

JMO : Integer full word variable containing the month at midpoint of accumulation.

JDA : Integer full word variable containing the day at midpoint of accumulation.

JHR : Integer full word variable containing the hour at midpoint of accumulation.

JMIN : Integer full word variable containing the minutes at midpoint of accumulation.

ISEC : Integer full word variable containing the seconds at midpoint of accumulation.

CONST : Floating point full word variable containing C.

ZNZ : Floating point full word variable containing  $n_0$ .

ERROR : Floating point full word variable containing the calculated error statistic.

## B. Variables and Arrays of Interest

IYR (1000) : Integer full word array containing the year at midpoint of accumulation read from the calibration cards.

IMO (1000) : Integer full word array containing the month at midpoint of accumulation read from the calibration cards.

IDA (1000) : Integer full word array containing the day at midpoint of accumulation read from the calibration cards.

IHR (1000) : Integer full word array containing the hour at midpoint of accumulation read from the calibration cards.

IMN (1000) : Integer full word array containing the minutes at midpoint of accumulation read from the calibration cards.

SEC (1000) : Floating point full word array containing the seconds at midpoint of accumulation read from the calibration cards.

SN1 (1000) : Floating point full word array containing the integrated channel number of the first peak in the calibration spectrum.

SN2 (1000) : Floating point full word array containing the integrated channel number of the second peak in the calibration spectrum.

SN3 (1000) : Floating point full word array containing the integrated channel number of the third peak in the calibration spectrum.

ICTS1 (1000) : Integer full word array containing the total counts surrounding the first peak.

ICTS2 (1000) : Integer full word array containing the total counts surrounding the second peak.

ICTS3 (1000) : Integer full word array containing the total counts surrounding the third peak.

C (1000) : Floating point full word array containing the calculated C for each calibration card read.

Z (1000) : Floating point full word array containing the calculated  $n_0$  for each calibration card read.

ERR (1000) : Floating point full word array containing the error statistic for each calibration card read.

MONTH (12) : Integer full word array containing the Julian day for the last day of each month of a normal year.

LMONTH (12) : Integer full word array containing the Julian day for the last day of each month of a leap year.

E1 : Floating point full word constant containing the energy associated with the first peak.

E2 : Floating point full word constant containing the energy associated with the second peak.

E3 : Floating point full word constant containing the energy associated with the third or sum peak.

SUM : Floating point full word variable containing the total  $n_0$  for all calibrations.

IHOLD : Integer full word variable containing the total number of calibration cards read.



ID : Integer full word variable containing the calibration card code (=200).

AVEZMZ : Floating point full word variable containing the average  $n_0$  for all cards read.

JUL : Integer full word variable containing the calculated Julian day.

IHRMN : Integer full word variable containing combined hours and minutes of day ( = HRS x 100 + minutes).

IRSLT : Integer full word variable containing the return code from a call to FPUT.

IRUNDA : Integer full word variable containing the run day after a call to FORTIM.

IRUNT : Integer full word variable containing the run time after a call to FORTIM.

LNECNT : Integer full word variable used as a line counter.

$n_0$  : Floating point full word containing the  $n_0$  to be used in the calculation of C(80.2).

## VI. Subroutine FORTIM (IRUNDA,IRUNT)

Program Language : IBM O/S 360 Assembler Language

Date Written : July 1969

Programmer : A. Buck

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments : 1) IRUNDA: output integer full word  
containing the date.  
  
2) IRUNT: output integer full word  
containing the time of day in  
hundredths of a second.

Subprograms Called : None

Description : Subroutine FORTIM uses the time macro  
to fetch the date and the time from  
the computer clock. The date and the  
time are only as accurate as the cor-  
responding information entered by the  
operator.

Example : FORTIM is called at 2.37 seconds after  
3 minutes after 1 o'clock, A.M. on  
February 10, 1970. When control is  
returned from subroutine FORTIM, IRUNDA  
contains 70041 in binary form and IRUNT  
contains 3780237 in binary form.

## VII. Subroutine FPUT (IDSN, ICMD, IRSLT, IOVEC)

Program Language : IBM O/S 360 Assembler Language

Date Written : February 4, 1971

Programmer : A. Buck

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments : 1) IDSN: a full word integer input parameter specifying the data set to which activity is directed. The integer stored at IDSN must be greater than zero but less than 21.

2) ICMD: a full word input integer variable containing a request type code which specifies the activity to be performed.

a) ICMD = 1 implies that a logical record is to be written.

b) ICMD = 2 implies the data set is to be ended. Once ended, the data set is unavailable to the calling program through the same data set number.

3) IRSLT: a full word output integer variable in which subroutine FPUT can place a return code as follows

a) IRSLT = -1 implies request was illegal for one of the reasons below:

i. IDSN contained a value less than 1 or greater than 20.

ii. ICMD contained a value less than 1 or greater than 2.

iii. A write request was made to a data set that had been ended.

iv. An endfile request was made to a data set either before the first write request or after the data set had been ended.

b) IRSLT = 0 implies request was executed normally.

c) IRSLT = 1 implies an I/O error occurred while writing a block. The data set was ended.

4) IOVEC: an input vector whose high order byte is the first byte to be written (when ICMD = 1) in a logical record.

The number of bytes written is specified by the LRECL subparameter on the DD statement describing the data set.

Subprograms Called : None

Description : Subroutine FPUT is a QSAM output subroutine to write logical records from an area defined by the calling program. This routine is specifically designed to allow a Fortran program to create without format control an extensive data set to which the DCB subparameter 'RECFM = FB' applies. For data sets to which 'RECFM = F' or RECFM = U applies, subroutine FTWRIT may be more easily used.

Subroutine FPUT can manage as many as 20 data sets in the same job step. As many as desired may be active simultaneously. Once the endfile command has been issued to a data set, subroutine FPUT will refuse to perform any further activity for that particular data set. The large number of data sets allows writing as many as 20 files to the same magnetic tape volume as a possible application.

DD Statements required...

If subroutine FPUT is called with IDSN=1 (or 2,..., or 20) the user must supply a DD Statement having the DD name FPUT01 (or FPUT02,..., or FPUT20). The DD Statement must include the DCB subparameters 'RECFM', 'LRECL', and 'BLKSIZE' but must omit 'DSORG' and 'MACRF'.

### VIII. Description of Input (CAL1)

The Pass-I calibration cards have the following format:

[illegible]

YEAR -----

MONTH

DAY

HOUR

MINUTES

SECONDS -----

Midpoint scantime.

CHANNEL (1)

Channel number of 1st peak channel

CHANNEL (2)

Channel number of 2nd peak channel

CHANNEL (3)

Channel number of 3rd peak channel

COUNTS (1)

Sum of counts around 1st peak channel

COUNTS (2)

Sum of counts around 2nd peak channel

COUNTS (3)

Sum of counts around 3rd peak channel

ID

= '200' for calibration cards

## IX. Description of Output (CAL1)

CAL1 output tape format is as follows:

(LRECL = 36, BLKSIZE = 1800)

YEAR	-----	}	Midpoint time
MONTH			
DAY			
HOUR			
MINUTES			
SECONDS	-----		

C

Calculated for each input card  
in the following manner:

$$w_i = \frac{N_i}{N_1 + N_2 + N_3}$$

$$C = \sum_i w_i \frac{E_i}{(n_i - n_0)^2}$$

$$E_1 = 1.7323$$

$$E_2 = 1.33248$$

$$E_3 = 2.525$$

$n_0$

The  $n_0$  that is read in the first  
input card - for our applications  
80.2 is used.

RUN DATE YEAR / DAY OF YEAR 73163 HRS / MIN 1617

SCAN TIME	PEAK CHANNELS AND COUNTS							CTS3	ERROR	
MID-POINT	C	AVE NO	INDIV NO	CH1	CH2	CH3	CTS1			CTS2
71 347 12 13 1218:20	0.4723E-04	78.3	84.0	77.3	87.9	150.9	1143	997	100	0.0023
71 347 12 13 1320:1	0.4655E-04	78.3	64.1	79.0	88.4	152.3	965	721	75	0.0096
71 347 12 13 1350:35	0.4698E-04	78.3	77.3	77.9	88.1	152.0	1153	980	76	0.0036
71 347 12 13 1458:10	0.4637E-04	78.3	76.2	79.0	89.2	152.5	944	778	72	0.0061
71 347 12 13 1522:44	0.4699E-04	78.3	77.3	77.9	88.1	151.2	1092	891	101	0.0038
71 347 12 13 1633:24	0.4617E-04	78.3	63.4	79.7	89.1	152.5	886	731	57	0.0134
71 347 12 13 1657:58	0.4695E-04	78.3	75.7	78.0	88.1	151.9	1127	934	110	0.0030
71 347 12 13 1805:33	0.4619E-04	78.3	66.5	79.6	89.2	153.0	919	811	82	0.0092
71 347 12 13 1830:7	0.4685E-04	78.3	84.9	77.9	88.6	152.4	1025	850	88	0.0047
71 348 12 14 1114:36	0.4352E-04	78.3	80.4	84.0	94.8	160.3	939	769	91	0.0031
71 348 12 14 1139:10	0.4406E-04	78.3	83.0	82.9	93.8	159.2	1133	996	103	0.0015
71 348 12 14 1314:24	0.4449E-04	78.3	82.3	82.1	92.9	158.9	1118	921	106	0.0074
71 348 12 14 1421:56	0.4443E-04	78.3	82.2	82.2	93.0	159.6	824	676	67	0.0119
71 348 12 14 1446:33	0.4495E-04	78.3	62.8	81.8	91.3	157.1	1191	883	85	0.0099
71 348 12 14 1554:7	0.4451E-04	78.3	82.3	82.1	92.9	158.1	904	772	73	0.0014
71 348 12 14 1618:42	0.4510E-04	78.3	83.4	81.0	91.8	156.7	1170	876	76	0.0029
71 348 12 14 1901:30	0.4484E-04	78.3	71.9	81.8	91.9	156.6	601	509	44	0.0061
71 348 12 14 1926:5	0.4537E-04	78.3	72.9	80.8	90.9	155.8	1115	926	110	0.0041
71 349 12 15 903:10	0.4367E-04	78.3	69.7	84.0	94.1	160.2	944	779	74	0.0056
71 349 12 15 1035:20	0.4366E-04	78.3	69.7	84.0	94.1	160.9	851	723	75	0.0076
71 349 12 15 1235:7	0.4403E-04	78.3	81.4	83.0	93.8	159.2	1232	999	85	0.0008
71 349 12 15 1410:21	0.4444E-04	78.3	63.4	82.7	92.3	159.5	1120	931	103	0.0146
71 349 12 15 1517:56	0.4443E-04	78.3	83.7	82.2	93.1	158.1	625	548	60	0.0020
71 349 12 15 1542:30	0.4475E-04	78.3	73.3	81.9	92.1	157.9	1172	974	87	0.0060
71 349 12 15 1650:5	0.4446E-04	78.3	82.2	82.2	93.0	157.5	747	604	55	0.0052
71 349 12 15 1714:39	0.4485E-04	78.3	73.5	81.7	91.9	157.7	1118	982	103	0.0065
71 349 12 15 1825:18	0.4478E-04	78.3	93.7	81.3	92.8	157.1	712	584	68	0.0075
71 349 12 15 1849:53	0.4523E-04	78.3	74.2	81.0	91.2	156.7	1161	966	93	0.0063
71 350 12 16 851:33	0.4482E-04	78.3	71.9	81.8	91.9	157.9	1132	939	87	0.0078
71 350 12 16 959:8	0.4452E-04	78.3	82.3	82.1	92.9	157.2	879	725	57	0.0065
71 350 12 16 1158:55	0.4452E-04	78.3	82.3	82.1	92.9	157.2	1157	905	94	0.0064
71 350 12 16 1331:5	0.4461E-04	78.3	82.5	81.9	92.7	157.8	1119	886	96	0.0015
71 350 12 16 1438:40	0.4470E-04	78.3	73.2	82.0	92.2	157.9	522	477	52	0.0052
71 350 12 16 1506:18	0.4520E-04	78.3	74.1	81.1	91.3	155.4	1165	927	76	0.0072
71 350 12 16 1638:27	0.4511E-04	78.3	83.4	81.0	91.8	156.1	1107	904	99	0.0031
71 350 12 16 1746:2	0.4464E-04	78.3	75.0	81.7	92.0	157.0	609	533	54	0.0029
71 350 12 16 1813:41	0.4565E-04	78.3	75.0	80.2	90.4	156.8	1114	885	102	0.0157
71 351 12 17 640:7	0.5082E-04	78.3	91.4	71.4	82.1	142.9	753	686	62	0.0067
71 351 12 17 815:21	0.4693E-04	78.3	75.7	78.0	88.1	152.7	1117	852	90	0.0086
71 351 12 17 1052:1	0.4550E-04	78.3	85.7	80.2	91.1	156.2	738	644	57	0.0076
71 351 12 17 1227:14	0.4532E-04	78.3	72.8	80.9	91.0	155.3	692	478	60	0.0061
71 351 12 17 1359:23	0.4554E-04	78.3	84.2	80.2	91.0	154.6	467	428	31	0.0060
71 351 12 17 1427:2	0.4562E-04	78.3	85.9	80.0	90.9	155.8	1156	882	91	0.0069
71 351 12 17 1534:37	0.4569E-04	78.3	73.5	80.2	90.3	155.7	553	423	52	0.0079
71 351 12 17 1841:13	0.4387E-04	78.3	60.7	83.9	93.4	159.4	603	491	68	0.0108
71 351 12 17 1908:52	0.4447E-04	78.3	80.6	82.2	92.9	158.9	1084	974	84	0.0068
71 352 12 18 603:9	0.5511E-04	78.3	74.1	65.9	75.1	133.2	801	694	76	0.0070
71 352 12 18 707:40	0.5006E-04	76.3	79.3	72.9	82.9	144.9	967	717	76	0.0046
71 352 12 18 735:18	0.4933E-04	78.3	81.2	74.0	84.2	145.9	1091	879	85	0.0015
71 352 12 18 839:49	0.4762E-04	78.3	76.8	76.9	87.0	148.5	896	760	69	0.0137
71 352 12 18 910:32	0.4743E-04	78.3	75.0	77.2	87.2	151.0	1015	849	90	0.0051
71 352 12 18 1015:2	0.4687E-04	78.3	84.9	77.9	88.6	151.4	918	680	64	0.0051
71 352 12 18 1042:41	0.4687E-04	78.3	75.6	78.1	88.2	152.9	1101	827	81	0.0091
71 352 12 18 1217:55	0.4671E-04	78.3	86.3	78.1	88.9	152.7	980	871	80	0.0048
71 352 12 18 1350:4	0.4672E-04	78.3	86.3	78.1	88.9	152.2	1165	914	82	0.0034

CALIBRATION LISTING - OUTPUT FROM CAL1

## Section 4

### CAL2

	<u>Page</u>
I. Introduction	4-1
II. Macro Systems Flowchart	4-3
III. Job Stream Schematic	4-4
IV. Job Control Language	4-5
V. CAL2 (Main Program)	4-6
VI. Subroutine CALNDR (USOC,IYR,IMO,IDA,IHR,IMN,SEC)	4-11
VII. Subroutine FGET (IDSN,IRSLT,IOVEC)	4-12
VIII. Subroutine FORTIM (IRUNDA,IRUNT)	4-14
IX. Subroutine SIMQ	4-15
X. Subroutine SOCENT (IYR,IMO,IDA,IHR,IMN,SEC,USOC)	4-17
XI. Description of Input	4-18
XII. Description of Output	4-20
XIII. Sample Output Listing	



## I. Introduction

The objective of CAL2 is to provide the information that will allow us to gain correct as much of the gain variable real pulse height spectra as possible into the gain corrected, known and nonvariable pulse height spectra output from PASS-III. These gain-corrected spectra can then be directly compared over any time period in a search for gamma ray sources.

To accomplish this, several bits of information are used. These are:

- A. Output of CALL - this provides known detector gain points at specific lines twice per orbit.
- B. Output of the VA Predict program - this provides the times and intensities of particular VA belt passages.
- C. Our acquired knowledge of typical detector operation during and after first turn on and specific intensity VA belt passage.

A typical day's operation might be as follows; after an eight hour turn off during the most intense VA belt passages, the instrument's HV are turned on. The central detector's gain drifts downward for several hours after this while coming to equilibrium. Sometimes during these first few hours there are several relatively minor step gain changes caused by slight "clipping" of the VA belt edges. After several hours the gain becomes constant in time and usually remains so for a period of 8 to 12 hours. At this time, the gain becomes affected as the spacecraft begins to "clip" the VA belts again. This effect increases until the VA belt intensity becomes great enough to require the turning off of the central detector's HV.

To handle these varying gain conditions, three quality indicators were developed.

Quality 1 - best data, peak position of each successive RS calibration should differ by one channel or less, no VA belt contamination.

Quality 2 - intermediate quality data, peak position from each successive RS calibration can differ by up to 3 - 4 channels but must do so smoothly in time, only very minor VA belt contamination allowed.

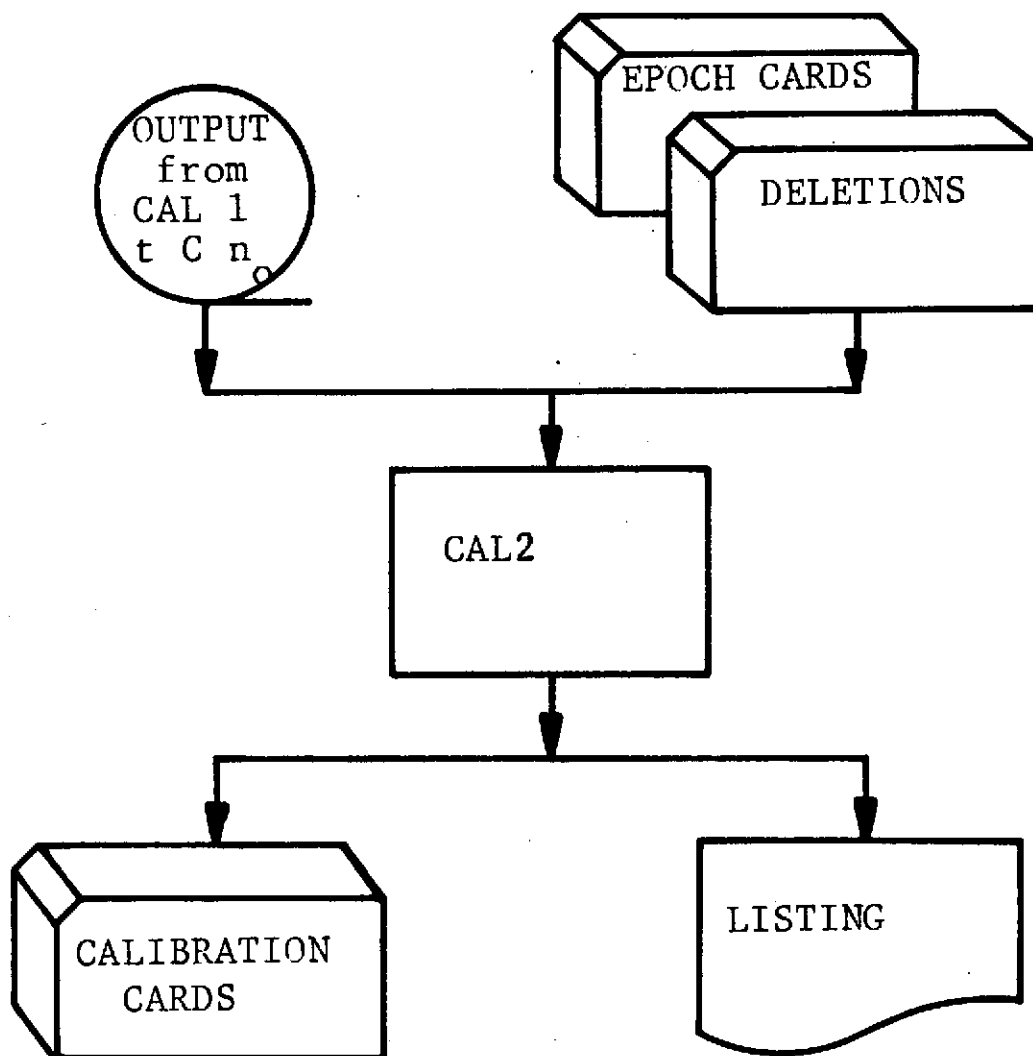
Quality 3 - all the rest of the data.

Hence we have high confidence that Quality 1 data has been gain corrected accurately, moderate confidence in Q2, and no confidence in Q3.

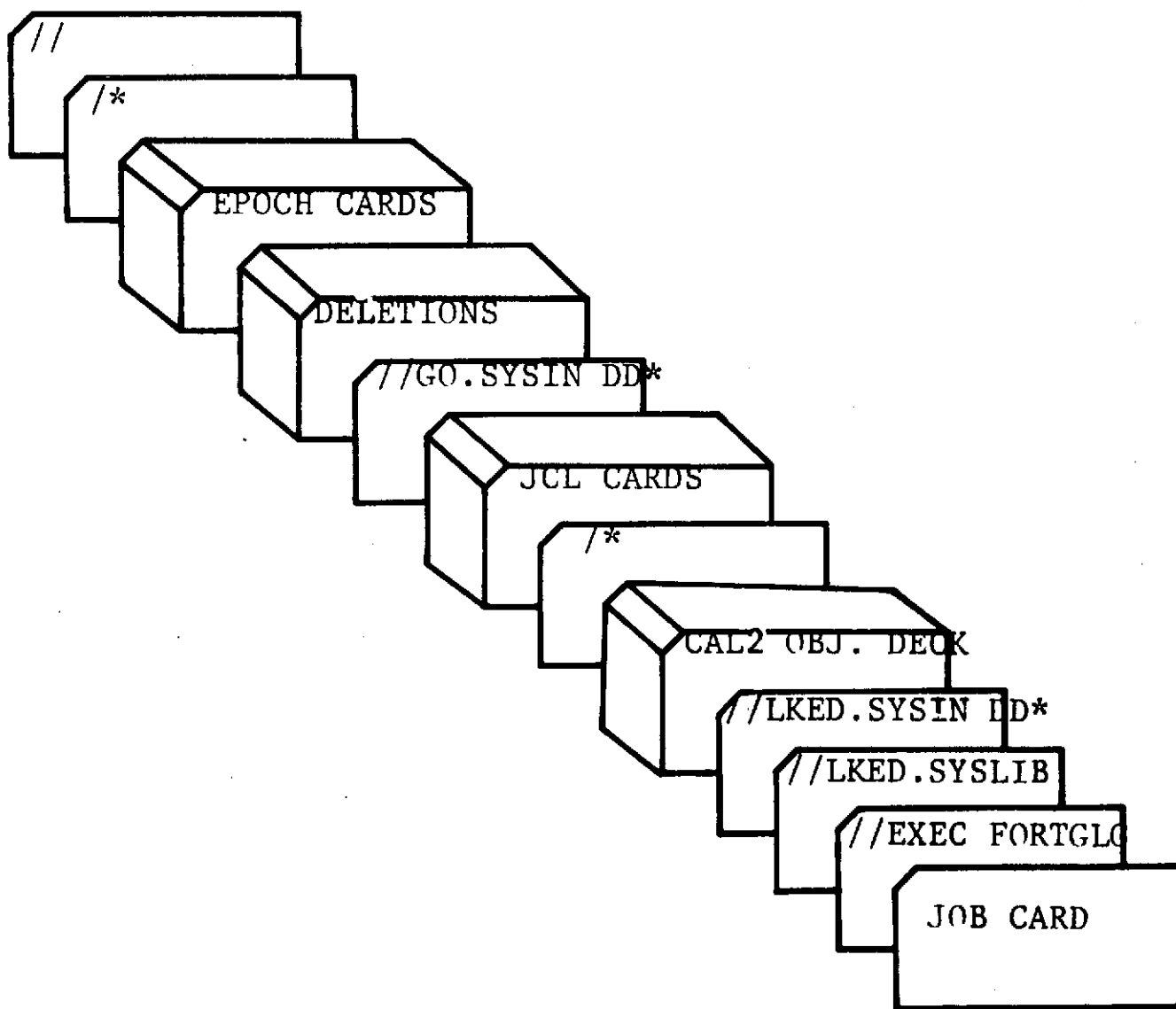
The boundary times of turn on and turn off, telemetry interruption, and quality changes are input values punched on epoch time cards.

All of this information combined with the RS CAL data from CAL1 forms the input to the computer operation portion of CAL2. The output of this is an analytical function giving the gain of the central detector as a function of time. This in turn forms one of the inputs of PASS-III so that all of the pulse height spectra can be gain corrected within the confidence levels specified by the quality indicator.

## II. Macro Systems Flowchart (CAL2)

C<sup>3</sup>

## III. Job Stream Schematic (CAL2)



## IV. JOB CONTROL LANGUAGE - CAL2

```
//      T548-12,,,,,,55),'S.CROTEAU,CAL2',CLASS=F,TIME=05,REGION=120K
//*LINECNT 62
//STEP1 EXEC  FORTGLG
//LKED.SYSLIB DD DSN=USER.T317.SUBLIB,DISP=SHR
//           DD DSN=USER.T548.SUBLIB,DISP=SHR
//           DD DSN=SYS1.FORTLIB,DISP=SHR
//           DD DSN=USER.SSPLIB,DISP=SHR
//LKED.SYSIN DD *
```

\*\*\* CAL2 OBJECT DECK \*\*\*

```
/*
//GO.FGET01 DD UNIT=2400,DISP=(OLD,KEEP),LABEL=(1,SL),
//          DSNAME=CAL1,DCB=(RECFM=FB,LRECL=36,BLKSIZE=1800),
//          VOLUME=SER=CC0922
//GO.SYSUDUMP DD SYSOUT=A
//GO.SYSIN DD *
```

\*\*\* DELETIONS AND EPOCH TIME CARDS \*\*\*

```
/*
//
```

## V. CAL2 (Main Program)

Program Language : Fortran IV, G-Level  
 Date Written : June 1972  
 Programmer : S. Croteau  
 Installation : University of New Hampshire  
 Computer : IBM-360-H50  
 Subprograms Called : FORTIM, SOCENT, CALNDR, SIMQ, FGET

This program reads in deletion cards, epoch time cards, and the output tape from CAL1. The deletion cards contain the times of tape records to be ignored and a '1' in column 80. The epoch time cards contain the epoch times, the quality indicator, and a '2' in column 80. The quality indicator is a number from 1 to 3, 1 being good quality and 3 being poor quality. This indicator is carried over on the Pass-III tape to all spectra falling within the range of this epoch. The calibration input tape records generated by CAL1 contain midpoint of accumulation time, C, and  $n_0$  (80.2). The deletions must always precede the epoch cards, both being in chronological order. The epoch time cards and the tape records merge so that a certain number of calibrations correspond to each epoch time.

CAL2 fits a gain function to data of a given epoch, T, in the form:

$$C(t) = q + r(t-T) + s(t-T)^2$$

Let N be the number of calibrations for one epoch.

$$\text{For } N = 1 \quad r = s = 0$$

$$q = c_1$$

$$\text{For } N = 2 \quad s = 0$$

$$c_1 = q + r(t_1 - T)$$

$$c_2 = q + r(t_2 - T)$$

$$r = \frac{c_2 - c_1}{t_2 - t_1}$$

$$q = c_1 - (c_2 - c_1) \left( \frac{t_1 - T}{t_2 - t_1} \right)$$

For  $N = 3$  using linear least squares fit:

$$\Delta = \sum_i (C - C_i)^2 \text{ with } s = 0$$

$$\frac{\partial \Delta}{\partial q} = \frac{\partial \Delta}{\partial r} = 0$$

$$\Delta = \sum (q + r\Delta t_i - C_i)^2 \text{ where } \Delta t_i = t_i - T$$

$$\frac{\partial \Delta}{\partial q} = \sum (q + r\Delta t_i - C_i) = 0$$

$$\frac{\partial \Delta}{\partial r} = \sum (q\Delta t_i + r(\Delta t_i)^2 - C_i\Delta t_i) = 0$$

$$3q + r\sum \Delta t_i = \sum C_i$$

$$q\sum \Delta t_i + r\sum (\Delta t_i)^2 = \sum C_i \Delta t_i$$

For  $N \geq 4$  using quadratic least squares:

$$\frac{\partial \Delta}{\partial q} = \frac{\partial \Delta}{\partial r} = \frac{\partial \Delta}{\partial s} = 0$$

$$N_q + r\sum \Delta t_i + s\sum (\Delta t_i)^2 = \sum C_i$$

$$q\sum \Delta t_i + r\sum (\Delta t_i)^2 + s\sum (\Delta t_i)^3 = \sum C_i \Delta t_i$$

$$q\sum (\Delta t_i)^2 + r\sum (\Delta t_i)^3 + s\sum (\Delta t_i)^4 = \sum C_i (\Delta t_i)^2$$

Subroutine SIMQ is used to solve for  $q$ ,  $r$ , and  $s$ .

Calibration cards are punched containing the epoch time,  $q$ ,  $r$ ,  $s$ ,  $n$ , the quality indicator, and an ID (300 or 301). A listing of the inputs and outputs is provided by the program.

The following is a description of some parameters used in the program:

#### A. Unlabelled Common

JYR : integer full word variable containing year at midpoint of accumulation read from input tape.

JMO : integer full word variable containing month at midpoint of accumulation read from input tape.

JDA : integer full word variable containing day at midpoint of accumulation read from input tape.

JHR : integer full word variable containing hour at midpoint of accumulation read from input tape.

JMN : integer full word variable containing minutes at midpoint of accumulation read from input tape.

ISEC : integer full word variable containing seconds at midpoint of accumulation read from input tape.

C : floating point full word variable containing the value of C for this calibration.

AVENO : floating point full word variable containing the average  $n_0$  for this calibration (80.2).

ERR : floating point full word variable containing the error statistic calculated in CAL1 for this calibration.

#### B. Variables and Arrays of Interest

A(9) : floating point full word array containing the coefficients stored column wise used as an argument to SIMQ.

B(3) : floating point full word array containing the original constants used in the call to SIMQ. These are replaced by final solution values.

CONST(1000) : floating point full word array containing the C value of the calibrations to be processed for an epoch time.

CTI : floating point full word variable containing the solution to the equation

$$CTI = (C - (q + r(t_1 - T) + s(t_1 - T)^2))/C$$

ERROR(1000) : floating point full word array containing the error statistic for the calibration records to be processed for a certain epoch.

FRACTN : floating point full word variable containing the solution to

$$\frac{t_1 - T}{t_2 - t_1}$$

ID1 : integer full word variable containing card code '300'.



ID2 : integer full word variable containing card code '301'.  
 IDELT : integer full word variable used as a subscript in the processing of deletions.  
 IEND : integer full word variable set to 1 to indicate the last epoch time card has been read.  
 IERROR : integer full word variable set to 1 to indicate that a calibration has been read which has no corresponding epoch time card.  
 IQLTY(2) : integer full word array containing the quality indicator for the two epoch times being compared with the calibration times on the input tape.  
 IQUAL(500) : integer full word array containing the quality indicator to be punched in the calibration cards.  
 IRSLT : integer full word variable containing the return code from the call to FGET.  
 IRUNDA : integer full word variable used as an argument in the call to FORTIM; contains the run day.  
 IRUNT : integer full word variable used as an argument in the FORTIM; contains the run time.  
 JEND : integer full word variable set to 1 to indicate the last tape record has been read.  
 KS : integer full word variable used as an output argument of SIMQ. 0 → normal operation  
 1 → singular set of equations.  
 KTIME : integer full word variable containing hours \* 100 + minutes.  
 NEQU : integer full word variable used to contain the number of equations to be solved by SIMQ (input argument).  
 QDEL : logical 8-bit indicator set to .TRUE. when a tape record is to be ignored.  
 Q(500) : floating point full word array containing the values of q to be punched.  
 R(500) : floating point full word array containing the values of r to be punched.  
 S(500) : floating point full word array containing the values of s to be punched.

T1 : floating point full word variable containing the solution to  $t_1 - T$ .

T2 : floating point full word variable containing the solution to  $t_2 - T$ .

T3 : floating point full word variable containing the solution to  $t_3 - T$ .

UDEL(500) : double word floating point array containing the deletion times in seconds of century.

UEPOCH(2) : double word floating point array containing the two epoch times (in seconds of century) that are being compared with the calibration tape.

UMID(1000) : double word floating point array containing the time in seconds of century of calibration records processed for an epoch.

UTIME : double word floating point variable containing the time in seconds of century of the tape record currently being processed.

UT(500) : double word floating point array containing the epoch times in seconds of century of the calibration cards to be punched.

ZNZ(1000) : floating point full word array containing the  $n_0$  read from the calibration records to be processed for a certain epoch time.

ZNZERO(500) : floating point full word array containing the average  $n_0$  to be punched.

## VI. Subroutine CALNDR (USOC,IYR,IMO,IDA,IHR,IMN,SEC)

Program Language : Fortran IV, G-Level

Date Written : February 1972

Programmer : S. Croteau

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments : 1) USOC: double precision input floating  
point variable containing seconds of  
century.

2) IYR: output integer full word con-  
taining year.

3) IMO: output integer full word con-  
taining month.

4) IDA: output integer full word con-  
taining day.

5) IHR: output integer full word con-  
taining hour.

6) IMN: output integer full word con-  
taining minutes.

7) SEC: output floating point full word  
containing seconds.

Subprograms Called : None

Description : This subroutine accepts seconds of cen-  
tury as input via the argument list and  
converts this time to calendar time --  
year, month, day, hours, minutes, and  
seconds. The input value in seconds of  
century is a 64-bit floating point word.  
The output variables are 32-bit integers  
except for the seconds which are stored  
in a 32-bit floating point word.

## VII. Subroutine FGET (IDSN,IRSLT,IOVEC)

Program Language : IBM O/S 360 Assembler Language

Date Written : February 4, 1971

Programmer : A. Buck

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments : 1) IDSN: a full word input integer containing the data set number to which the call is directed. (IDSN must be 1, 2, or 3).

2) IRSLT: a full word output integer in which subroutine FGET may place an activity-result code as follows...

a) IRSLT = -1 if the requested activity was invalid. This code is returned if either of the following occur:

i. IDSN contains a number less than 1 or greater than 3.

ii. Data set has been previously closed.

b) IRSLT = 0 if data is read and no I/O error is detected.

c) IRSLT = 1 if an I/O error occurred on the block containing the data placed in IOVEC. The calling program has the responsibility for accepting or rejecting the data. In addition, the calling program is responsible for proper reaction to the remaining records in the block for which IRSLT will be 0.

d) IRSLT = 2 if end-of-file was detected. The contents of IOVEC are not changed. The data set is closed and can not be used again by subroutine FGET with the same value of IDSN.

3) IOVEC: an output vector in which FGET can place the record. The actual variable used for IOVEC may be integer, real, logical, or complex. The first leftmost byte of the data record is placed in the high-order byte of IOVEC when reading occurs. The number of bytes placed in IOVEC is specified by the LRECL subparameter of the DD Statement.

Subprograms Called : None

Description : This subroutine is a QSAM input subroutine to read logical records into an area defined by the calling program. This routine is specifically designed to supply a Fortran calling program with logical records from a data set to which the DCB subparameter 'RECFM = FB' applies. For data sets to which 'RECFM = F' or 'RECFM = U' applies, subroutine FTREAD may be more easily used. "RECFM = V" is specifically prohibited.

Subroutine FGET can manage as many as 3 data sets during the same job step. The data sets may be open (active) simultaneously provided no two data sets are on the same magnetic tape.

DD Statements required...

If FGET is to be called with IDSN = 1 (or 2 or 3), then the user must describe the data set by a DD Statement having the DD name FGET01 (or FGET02 or FGET03). The DD Statement must include the DCB subparameters 'RECFM', 'LRECL', and 'BLKSIZE' but must omit 'DSORG' or 'MACRF'.

## VIII. Subroutine FORTIM (IRUNDA,IRUNT)

Program Language : IBM O/S 360 Assembler Language

Date Written : July, 1969

Programmer : A. Buck

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments : 1) IRUNDA: output integer full word  
containing the date.

2) IRUNT: output integer full word  
containing the time of day in hundredths  
of a second.

Subprograms Called : None

Description : Subroutine FORTIM uses the time macro  
to fetch the date and the time from  
the computer clock. The date and the  
time are only as accurate as the cor-  
responding information entered by the  
operator.

Example : FORTIM is called at 2.37 seconds after  
3 minutes after 1 o'clock A.M. on  
February 10, 1970. When control is  
returned from subroutine FORTIM, IRUNDA  
contains 70041 in binary form and IRUNT  
contains 3780237 in binary form.

## IX. Subroutine SIMQ (A,B,N,KS)

Program Language : Fortran IV, G-Level  
 Date Written :  
 Programmer :  
 Installation : University of New Hampshire  
                     Space Science Center  
 Computer : IBM-360-H50  
 Dummy Arguments : (A,B,N,KS)

A     Matrix of coefficients stored  
       columnwise. These are destroyed  
       in the computation. The size of  
       matrix A is N by N.

B     Vector of original constants  
       (length N). These are replaced  
       by final solution values, vector X.

N     Number of equations and variables.  
       N must be .GT. 1.

KS    Output digit  
       0 for a normal solution  
       1 for a singular set of equations

Purpose : Obtain solution of a set of simultaneous  
           linear equations,  $AX = B$ .

Subprograms Called : None

Description : This subroutine is found on page 178 of  
               System/360 Scientific Subroutine Package  
               published by IBM. The following is the  
               description of the program found on that  
               page.

Matrix A must be general.

If matrix is singular, solution values all  
 meaningless. An alternative solution may  
 be obtained by using matrix inversion  
 (MINV) and matrix product (GMPRD).

Method of solution is by elimination using largest pivotal divisor. Each stage of elimination consists of interchanging rows when necessary to avoid division by zero or small elements.

The forward solution to obtain variable N is done in N stages. The back solution for the other variable is calculated by successive substitutions. Final solution values are developed in vector B, with variable 1 in B(1), variable 2 in B(2),....., variable N in B(N). If no pivot can be found exceeding a tolerance of 0.0, the matrix is considered singular and KS is set to 1. This tolerance can be modified by replacing the first statement.



## X. Subroutine SOCENT (IYR,IMO,IDA,IHR,IMN,SEC,USOC)

Program Language : Fortran IV, G-Level

Date Written : February, 1972.

Programmer : S. Croteau

Installation : University of New Hampshire

Computer : IBM-360-H50

Dummy Arguments : 1) IYR: input integer fullword containing  
year (2 digits).

2) IMO: input integer fullword containing  
month.

3) IDA: input integer fullword containing  
day.

4) IHR: input integer fullword containing  
hour.

5) IMN: input integer fullword containing  
minutes.

6) SEC: input floating point variable con-  
taining seconds.

7) USOC: doubleword floating point output  
variable containing seconds of  
century.

Subprograms Called : None

Description : This subroutine converts calendar time  
to seconds of century.

## XI. Description of Input (CAL2)

### A. Deletion

[illegible]

YEAR

Year of deletion

MON

Month of deletion

DA

Day of deletion

HR

Hour of deletion

MIN

Minute of deletion

SEC

Second of deletion

ID

'1' for deletion

B. EPOCH time card

[illegible]

YEAR

Year of epoch time

MON

Month of epoch time

DA

Day of epoch time

HR

Hour of epoch time

MIN

Minute of epoch time

SEC

Second of epoch time

QUAL

Quality indicator

'1' = Good

'2' = Fair

'3' = Poor

ID

= '2' for epoch time card

## XII. Description of Output (CAL2)

[illegible][illegible]

YR	Year of epoch time
MO	Month of epoch time
DA	Day of epoch time
HR	Hour of epoch time
MIN	Minute of epoch time
SEC	Second of epoch time
QUAL	Quality indicator
	'1' = Good '2' = Fair '3' = Poor
q	This is the value of $C_0$
r	This is the value of $\frac{dC}{dt}$
$\bar{n}_0$	Average $n_0$ (80.2)
s	This is a second derivative. It is not equal to zero only if there are more than three cali- brations for the same epoch time; it is calculated using three equations.
ID	= '300' for calibration card type 1 = '301' for calibration card type 2

RUN DATE YEAR / DAY OF YEAR 73163 HRS / MIN 1738

SCAN TIME	MID-POINT OF ACCUM.	C	(C-C(T))/C	ERROR	Q	R	S	AVE NO	EPOCH TIME	QLTY
71 12 8 11 53 37.0	0.4497E-04	-0.4011E-02	0.9017E-02							
71 12 8 12 24 20.0	0.4544E-04	0.2795E-02	0.1178E-01							
71 12 8 13 25 46.0	0.4571E-04	0.1507E-02	0.4270E-02							
71 12 8 13 56 30.0	0.4586E-04	0.1521E-02	0.9702E-02							
71 12 8 14 57 56.0	0.4611E-04	0.4494E-03	0.3845E-02							
71 12 8 15 31 43.0	0.4624E-04	-0.2345E-03	0.3881E-02							
71 12 8 16 33 9.0	0.4642E-04	-0.2185E-02	0.7369E-02							
71 12 8 18 5 18.0	0.4688E-04	-0.6614E-03	0.6788E-02							
71 12 8 19 40 32.0	0.4725E-04	-0.7383E-03	0.7749E-02							
71 12 8 20 11 15.0	0.4747E-04	0.1523E-02	0.5255E-02							
		0.4486E-04	0.9384E-10	-0.5172E-15	80.2	71 12 8 11 0	1			
71 12 9 11 14 21.0	0.4440E-04	-0.2925E-02	0.1387E-01							
71 12 9 11 45 4.0	0.4473E-04	0.1388E-02	0.2961E-02							
71 12 9 12 46 30.0	0.4500E-04	0.1086E-02	0.2929E-02							
71 12 9 13 17 13.0	0.4509E-04	0.3024E-03	0.3544E-02							
71 12 9 14 21 44.0	0.4529E-04	-0.1248E-02	0.4593E-02							
71 12 9 14 52 26.0	0.4563E-04	0.3542E-02	0.6956E-02							
71 12 9 15 53 53.0	0.4578E-04	0.1573E-02	0.5476E-02							
71 12 9 17 29 6.0	0.4586E-04	-0.4299E-02	0.1538E-01							
71 12 9 20 36 29.0	0.4644E-04	-0.4583E-02	0.3052E-02							
71 12 9 21 4 8.0	0.4697E-04	0.5079E-02	0.3220E-02							
		0.4418E-04	0.8153E-10	-0.4382E-15	80.2	71 12 9 10 0	1			
71 12 10 12 10 18.0	0.4474E-04	-0.2967E-02	0.1279E-01							
71 12 10 12 37 57.0	0.4522E-04	0.5796E-02	0.1241E-01							
71 12 10 13 42 27.0	0.4480E-04	-0.7935E-02	0.6957E-02							
71 12 10 14 13 10.0	0.4550E-04	0.5568E-02	0.5470E-02							
71 12 10 15 17 41.0	0.4524E-04	-0.4551E-02	0.9189E-02							
71 12 10 15 45 19.0	0.4580E-04	0.5685E-02	0.3242E-02							
71 12 10 16 49 50.0	0.4566E-04	-0.1296E-02	0.1575E-01							
71 12 10 13 52 42.0	0.4614E-04	0.2652E-03	0.8698E-02							
71 12 10 19 57 13.0	0.4614E-04	-0.4193E-02	0.4856E-02							
71 12 10 20 24 51.0	0.4658E-04	0.3428E-02	0.1066E-01							
		0.4467E-04	0.4987E-10	0.5773E-16	80.2	71 12 10 11 0	1			
71 12 11 11 34 6.0	0.4283E-04	0.6431E-03	0.1165E-01							
71 12 11 11 58 41.0	0.4304E-04	-0.2850E-03	0.5065E-02							
71 12 11 13 6 16.0	0.4344E-04	-0.4916E-02	0.9291E-02							
71 12 11 13 33 54.0	0.4418E-04	0.6975E-02	0.9347E-02							
71 12 11 14 41 29.0	0.4397E-04	-0.7842E-02	0.4773E-02							
71 12 11 15 6 3.0	0.4472E-04	0.5969E-02	0.5161E-02							
71 12 11 16 13 38.0	0.4471E-04	-0.7083E-03	0.4305E-02							
		0.4172E-04	0.2120E-09	-0.3443E-14	80.2	71 12 11 10 0	1			
71 12 11 16 41 17.0	0.4547E-04	-0.8821E-03	0.7439E-02							
71 12 11 17 45 47.0	0.4504E-04	0.2274E-02	0.3223E-02							

CALIBRATION LISTING - OUTPUT FROM CAL2

## Section 5

### Pass-III

	<u>Page</u>
I. Introduction	5-1
II. Macro Systems Flowchart	5-3
III. Job Stream Schematic	5-4
IV. Job Control Language	5-5
V. Pass-III Main Program (OSOP301A)	5-6
VI. Subroutine CALNDR (USOC,IYR,IMO,IDA,IHR,IMN,SEC)	5-16
VII. Subroutine CELFAR (RA,DEC,SATRA,SATDEC,THETA,PHI)	5-17
VIII. Subroutine CROSS (A,B,AXBN)	5-18
IX. Subroutine FEEDER	5-19
X. Subroutine FORTIM (IRUNDA,IRUNT)	5-20
XI. Subroutine FPUT (IDSN,ICMD,IRSLT,IOVEC)	5-21
XII. Subroutine GAINB (CHAN,MONE1,MTW02)	5-23
XIII. Entry HPARM	5-25
XIV. Entry INTPRM	5-26
XV. Subroutine PARM	5-27

XVI.	Subroutine RECSPH (THETA,PHI,V)	5-28
XVII.	Subroutine ROTATE (N,AA,V,VP)	5-29
XVIII.	Subroutine SOCENT (IYR,IMO,IDA,IHR,IMN,SEC,USOC)	5-30
XIX.	Subroutine SPHREC (THETA,PHI,V)	5-31
XX.	Subroutine VABELT (GLON,GLAT,ALT,FLUX)	5-32
XXI.	Subroutine VCR (GLON,GLAT,PC)	5-33
XXII.	Description of Input	5-34
XXIII.	Description of Output	5-42
XXIV.	Sample Output Listing	



## Pass-III

### I. Introduction

Pass-III reads in as input a control card, Van Allen cards, good time cards, calibration cards, and the Experimenter data tape.

The control card contains the parameters needed for the initial call to FEEDER.

The Van Allen cards contain the Van Allen data accumulated in the previous Pass-III production run in the form of calendar time, peak flux, and integrated flux. Each time a new file of data is read, the Van Allen data values are calculated from the orbit data records found at the beginning of the file. If no Van Allen cards are entered, the program initializes the Van Allen data array to zeroes and begins accumulating VA data from the first file read on tape 1 of the current run.

The good time cards (generated by Pass-I) contain the beginning and end times of good complete scans of data (solar or background or both). This is data in which the sync word was present, in which there was no missing data (no time gaps), and which was accumulated when the central detector was on.

The calibration cards (generated by CAL2) contain the epoch time, data quality indicator, and  $q, r, s$ , and  $n_0$  used to gain correct the spectra.

Pass-III stores the values from each of these cards in arrays. The calendar times are converted to seconds of century. The first good time is then read and the Experimenter tape is searched until the corresponding time to the nearest second is found. Processing begins at background or solar quadrant depending on what is specified in the good time card (BO or SO). Pass-III processes six major frames at a time (equivalent to one solar quadrant or one background quadrant) until the first end time is reached. Processing stops at background or solar depending on what is requested in the good time card (BK or BO). Pass-III then reads the next begin time and continues processing in the same manner as just described until the last end time is reached.

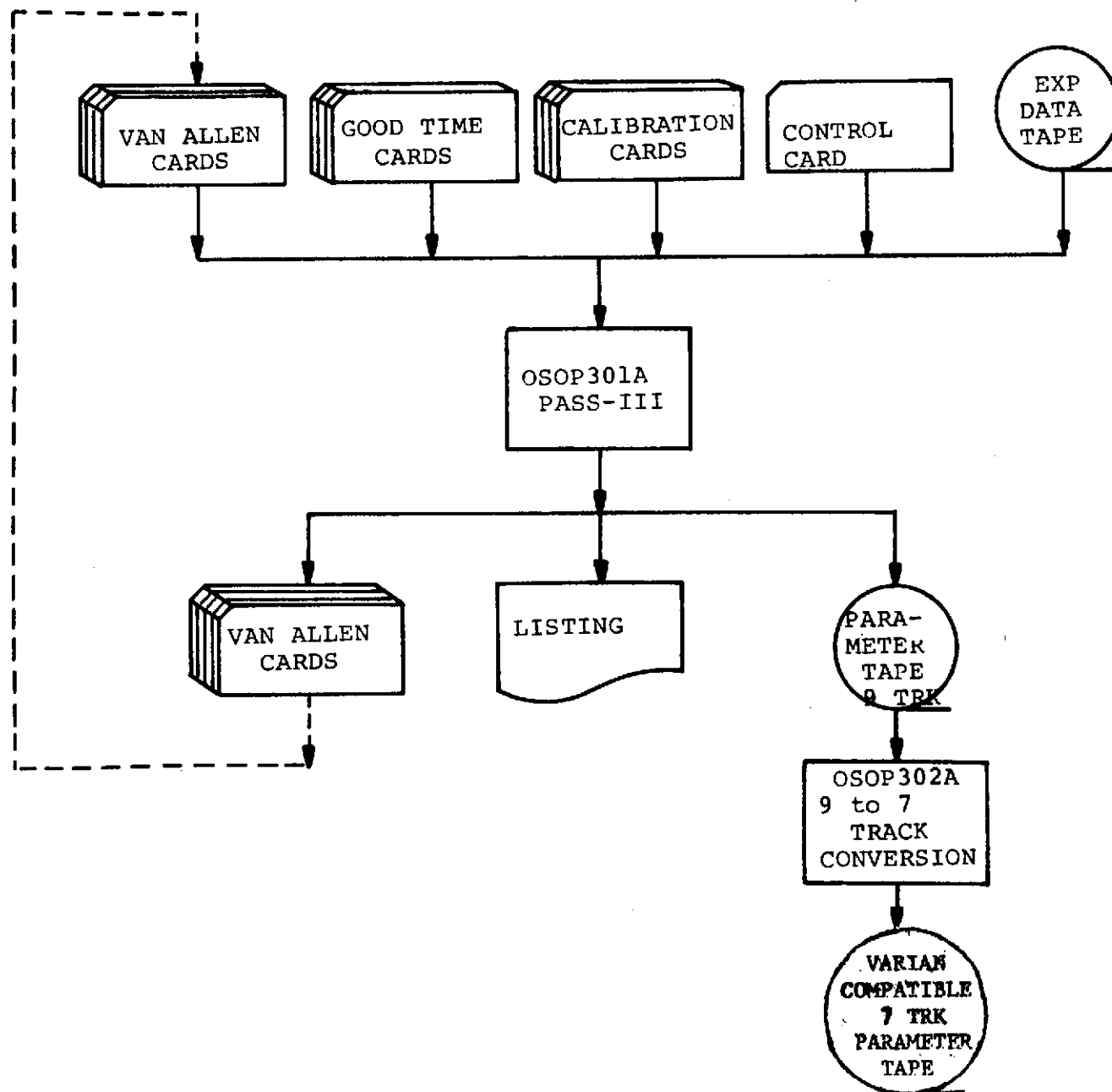
For each quadrant accepted as good data, orbit and attitudinal parameters are calculated and the gamma spectrum is gain shifted. The status word is realigned with the data spectrum and interpreted. Various flags are set to indicate the condition of the data processed, e.g. the presence of bit errors, dead time fold over, I/O errors, etc.

The output of Pass-III is a tape containing the parameters, calculated angles, gain-shifted spectrum, interpreted status, and flags for each half-scan or six major frames processed. This tape has a record length and block size of 3540 bytes.

A short listing is produced by Pass-III consisting of the first good time and last end time, tape volume serial numbers of tapes read, Decomm run numbers for each tape, the run date, and various warnings when unusual problems occur.

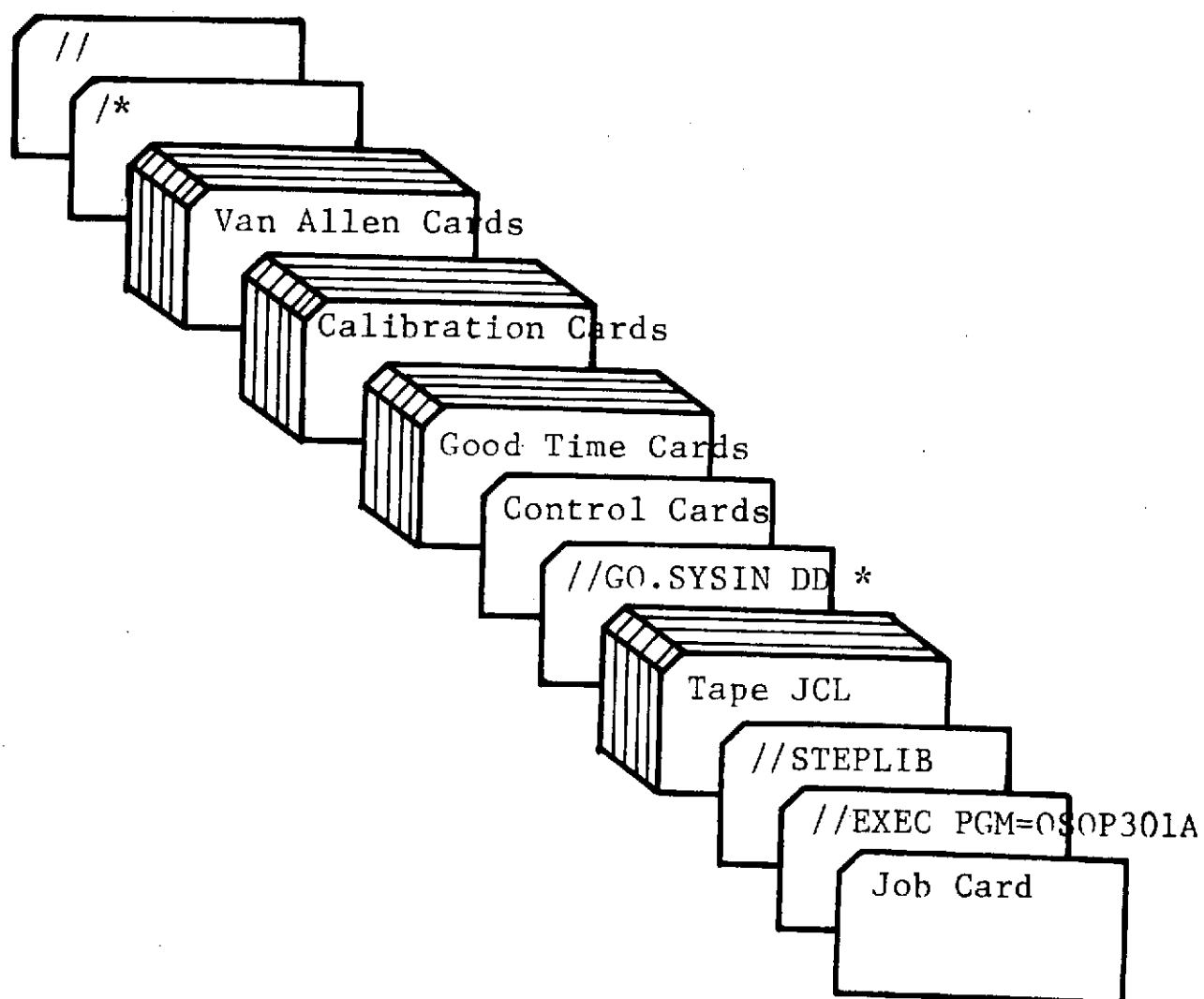
## II MACRO SYSTEMS FLOWCHART

PASS - III



## III. JOB STREAM SCHEMATIC

PASS-III



## IV. JOB CONTROL LANGUAGE - PASS-III

```

//          T548-13,,10),'S.CROTEAU,PASS-III',CLASS=F,TIME=15,REGION=160K
//STEP1 EXEC PGM=OSOP301A,TIME=15
//STEPLIB DD DSN=USER.T548.SUBLIB,DISP=SHR
//DECOM1 DD UNIT=TAPE7,DISP=OLD,LABEL=(11,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD329
//DECOM2 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD307
//DECOM3 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD308
//DECOM4 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD346
//DECOM5 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD347
//DECOM6 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD329
//DECOM7 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD295
//DECOM8 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD306
//DECOM9 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD283
//DECOM10 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,DEN=1,TRTCH=C),
//          VOLUME=SER=EXD295
//DECOM11 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,TRTCH=C),
//          VOLUME=SER=EXD296
//DECOM12 DD UNIT=AFF=DECOM1,DISP=OLD,LABEL=(1,NL,,IN),
//          DCB=(BLKSIZE=1000,TRTCH=C),
//          VOLUME=SER=EXD297
//FPUT01 DD UNIT=2400,DISP=(NEW,KEEP),LABEL=(1,SL),
//          DSN=PASSIII,DCB=(RECFM=F,LRECL=3540,BLKSIZE=3540),
//          VOLUME=SER=PRM057
//FT06F001 DD SYSOUT=A,DCB=(RECFM=UA,BLKSIZE=133)
//MSGDD DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//FT07F001 DD SYSOUT=B,DCB=(RECFM=F,BLKSIZE=80)
//FT05F001 DD *
1 1 1 1 1 1 1 10 15

```

\*\*\* GOOD TIME CARDS, CALIBRATION CARDS, VAN ALLEN CARDS \*\*\*

```

/*
//

```

## V. Pass-III Main Program (OSOP301A)

Program Language : Fortran IV, G-Level  
Date Written : April 1972  
Programmer : S. Croteau  
Installation : University of New Hampshire  
Computer : IBM-360-H50  
Subprograms Called : INTPRM, SOCENT, HPARM, FEEDER,  
CALNDR, PARM, FORTIM, GAINB, FPUT

This program prepares an area in common storage containing all the necessary parameters for one tape record. The following operations are performed by the program in the preparation of the parameter tape record:

- A. Reads in the data cards (good time cards, calibration cards, Van Allen cards) and stores the data values in arrays.
- B. Reads in the Experimenter data tape using subprogram FEEDER.
- C. Converts the data times from read-out time to accumulation time in seconds of century.
- D. Matches good times with data and begins processing at proper scan (background or solar).
- E. Call HPARM at the start of each file read to fetch Van Allen parameters (time, peak flux, and integrated flux). The program stores these values in arrays. The first 14 values should always have times less than the data time being processed. The fifteenth should always be later.
- F. Realignment the status so that it corresponds to the main frame data. In order to do this the program saves the status and the interpreted status bits from FEEDER and uses them with the main frame data read out three minutes later.
- G. Calls PARM for each three-minute scan to calculate orbit and attitudinal parameters.
- H. Calculates the Julian date.

- I. Calculates the satellite-earth-sun angle and the horizon angle in the following way:

$$\text{Satellite-earth-sun angle} = (\text{Satellite RA}) - (\text{Sun RA})$$

(where RA right ascension)

$$\text{Horizon angle} = \tan^{-1} \left( \frac{\text{radius earth}}{\text{radius earth} + \text{alt}} \right) * \frac{180}{\pi}$$

- J. Calculates the total x-ray counts in channel 1 from the interpreted digital subcom source values.
- K. Calls FORTIM to fetch run date.
- L. Checks the major frames for I/O errors and sets a flag.
- M. Calls GAINB to gain correct the spectrum.
- N. Counts the number of flagged MF data words in a spectrum and stores the value.
- O. Checks the spectrum for bit errors and sets a flag.
- P. Calculates the live time and dead time.
- Q. Determines the existence of dead time fold over, adjusts the dead time if fold over exists, and sets a flag.
- R. Computes rates surrounding  $\text{Co}^{60}$  leakage peaks for each spectrum

$$\frac{\text{Ch. 151-155} + \text{Ch. 161-165}}{\text{Ch. 144-148} + \text{Ch. 168-172}}$$

- S. Calculates the average rate ( $\text{Co}^{60}$ ) which should be an average of all continuous spectra, background or solar, between calibrations.
- T. Calls FPUT to write a record.
- U. Reads the next good time card.
- V. Compares the data time to the next calibration time. If the data time is greater, the program uses this calibration to make the gain correction.
- W. At end of job, the program punches the first 14 positions of the Van Allen data tables. These will be used as input into the next Pass-III run.

The following is a description of common storage and of the variables and arrays that are of major interest within the program. All variable names beginning with a U are double precision, those beginning with an H are integer half-words, and those beginning with a Q are logical bytes (8 bits).

- A. COMMON/HEADER/  
COMMON/ORBIT/  
COMMON/DECIDE/  
COMMON/SCANS/  
COMMON/INTRUP/      These areas are described in Section 1 of this report.
- B. COMMON/PARMLS/      This area is filled by each call to PARM.
- UVATIM(4)      : double precision floating point array containing the times of peak flux encountered.
- PC      : floating point full word variable containing rigidity.
- GLON      : floating point full word variable containing longitude.
- GLAT      : floating point full word variable containing latitude.
- ALT      : floating point full word variable containing altitude.
- SQRA      : floating point full word variable containing sun quadrant right ascension.
- SQDEC      : floating point full word variable containing sun quadrant declination.
- BQRA      : floating point full word variable containing background quadrant right ascension.
- BQDEC      : floating point full word variable containing background quadrant declination.
- SPNRA      : floating point full word variable containing spin right ascension.
- SPNDEC      : floating point full word variable containing spin declination.
- SQTHE      : floating point full word variable containing theta sun quadrant.



SQPHI	: floating point full word variable containing phi sun quadrant.
BQTHE	: floating point full word variable containing theta background quadrant.
BQPHI	: floating point full word variable containing phi background quadrant.
SPNTHE	: floating point full word variable containing theta spin axis.
SPNPHI	: floating point full word variable containing phi spin axis.
SATRA	: floating point full word variable containing satellite right ascension.
SATDEC	: floating point full word variable containing satellite declination.
ALAT	: floating point full word variable used by another program calling PARM.
VAPEAK(4)	: floating point full word array containing the peak flux.
VAFLUX(4)	: floating point full word array containing the integrated flux.
IDTAPE	: integer full word variable set aside to contain tape identification.
IDFILE	: integer full word variable set aside to contain file identification.
NORB	: integer full word variable used by another program calling PARM.
NVAPAS	: integer full word variable containing the number of peak fluxes encountered.
NVAPRN	: integer full word variable used by another program calling PARM.
BGAU	: floating point full word variable containing the magnetic field strength B (Gauss).
ERMCL	: floating point full word variable containing the McIllwain L parameter.
SUNRA	: floating point full word variable containing the sun right ascension.

SUNDEC : floating point full word variable containing the sun declination.

QVUPDT : logical 8-bit flag which remained unused.

QVSTRT : logical 8-bit flag which remained unused.

QATT : logical 8-bit indicator set to .TRUE. if no attitude data available.

QORB : logical 8-bit flag set to .TRUE. to indicate no orbit data available.

QPM : logical 8-bit flag set to .TRUE. if it was not possible to calculate sun RA or sun DEC.

QXTRA(3) : logical 8-bit flags used as filler.

C. COMMON/RECORD/ See Part XXIII of this "Description of Output."

D. COMMON/GAINLS/ The parameters in this storage area are input values into subroutine GAINB.

CVAL : floating point full word variable containing the value of  $q + r(t-T) + s(t-T)^2$ .

SK : floating point full word variable containing the standard C of .00005 (or  $5 \times 10^{-4}$ ).

NNAUGT : floating point full word variable containing  $n_0$ .

CHCTS(377) : floating point full word array containing the raw data spectrum prior to gain correction.

E. Variables and Arrays of Interest (not in common).

AMXFLX(30) : floating point full word array containing the most recent entries of Van Allen peak flux.

BKLR : floating point full word variable containing the sum of the lower range background X-ray interpreted values.

BKQDT : floating point full word variable containing the sum of background quadrant time.

BRATES(5) : floating point full word array of background  $\text{Co}^{60}$  rates to be summed.

BRATE1 : floating point full word variable used as a hold area for background X-ray rate (background quadrant).

BRATE2 : floating point full word variable used as a hold area for solar X-ray rate (background quadrant).

CNSTNT : floating point full word variable into which q or peak flux are read from the input cards.

CONST(200) : floating point full word array into which are stored all values of q from the calibration cards.

DERIV1(200) : floating point full word array into which are stored all values of r from the calibration cards.

DERIV2(200) : floating point full word array into which are stored all values of s from the calibration cards.

DERVT : floating point full word variable into which r or s or the integrated flux are read from input cards.

HBNUM : half-word integer variable containing the number of background entries in BRATES.

HB1 : half-word integer variable used for storing a background status word.

HDAYHD : half-word integer variable holding day-night status:  
1 → day  
2 → night

HFILE : half-word integer variable containing the file count of the current file.

HQUAL(200) : half-word integer array into which are stored all quality indicators from the calibration cards.

HSNUM : half-word integer variable containing the number of solar entries in SRATES.

HS1 : half-word integer variable used to store a solar status word.

IASOP : integer full word variable specifying the analog subcom data option desired:  
0 → omission of analog subcom data  
1 → analog subcom data to be processed

IATTOP : integer full word variable specifying the attitude data option desired:  
0 → omission of attitude data  
1 → attitude data to be processed

IBEG : double word floating point variable containing the begin time (in seconds of century) being dealt with.

ICAL : integer full word variable containing the total number of calibration entries.

ICHG : integer full word variable used in the determination of the presence or absence of a bit error in the spectrum.

IDSOP : integer full word variable specifying the digital subcom data option desired:  
0 → omission of digital subcom data  
1 → digital subcom data to be processed

ID1 : integer full word variable into which the card code is read.

IEND : double word floating point variable containing the end time (in seconds of century) being dealt with.

IF1 : integer full word variables used in the determination of the presence or absence of bit errors in the gamma spectrum: IF1 = 1; IF2 = 10; IF3 = 15.

IF2

IF3

IGOOD : integer full word variable containing the total number of good time cards read.

IORBOP : integer full word variable specifying the orbit data option desired:  
0 → no orbit data  
1 → regular orbit data items desired  
2 → all orbit data items desired

IREC : integer full word variable containing the number of output records written.

IRSLT : see Part XI of this section.

IRUNDA : see Part X of this section.

IRUNT : see Part X of this section.

ISBCAL : integer full word variable used as an index to calibration arrays.

ISBGDT : integer full word variable used as an index to the good time card arrays.

ISCANT : double word floating point variable containing the time of the current scan in seconds of century.

ISOBK(3) : integer full word array containing alphameric representations of background (BK), solar (SO), and both (BO).

ITYPE1(200) : integer full word array containing the scan type corresponding to begin time.

ITYPE2(200) : integer full word array containing the scan type corresponding to the end time.

ITYP1 : integer full word variable containing BO or SO to indicate whether the begin time starts with both quadrants or solar only.

ITYP2 : integer full word variable containing BK or BO to indicate whether the end time ends with both scans or background only.

LMONTH(12) : integer full word array containing the Julian day of the last day of each month of a leap-year.

MFDOP : integer full word variable specifying the main frame data option desired:  
 0 → no main frame data to be processed  
 1 → all main frame data to be processed  
 2 → PHA channel counts to be ignored

MONTH(12) : integer full word array containing the Julian day of the last day of each month of a regular year.

NTAPES : integer full word variable containing the number of tapes to be read.

QB(10) : logical array holding the interpreted status bits for a background spectrum.

QMESS : logical 8-bit flag controlling a message printed at abnormal run termination by the program.

QS(10) : logical array holding the interpreted status bits for a solar spectrum.

QSEEK : logical 8-bit flag set to .TRUE. if no status word has been saved.

QSOLAR : logical 8-bit flag set to .TRUE. when PARM has to be called while processing a solar scan (usually PARM is called only when processing a background scan).

QX(10) : logical array used as an intermediate area in the transfer of data from QS or QB to QPSTAT or vice versa.

RATE(2) : floating point full word array containing the background and solar X-ray rates for channel 1.

SAVED1 : double word floating point variable containing the sum of channels 151-155 and 161-165.

SAVED2 : double word floating point variable containing the sum of channels 144-148 and 168-172.

SOLLR : floating point full word variable containing the sum of the lower range solar interpreted values.

SOLQDT : floating point full word variable containing the sum for solar quadrant time.

SRATES(50) : floating point full word array of solar Co<sup>60</sup> rates to be summed.

SRATE1 : floating point full word variable used as a hold area for background X-ray rate (solar quadrant).

SRATE2 : floating point full word variable used as a hold area for solar X-ray rate (solar quadrant).

TCLOCK : floating point full word variable containing the clock time.

TEG(30) : floating point full word array containing the most recent entries of integrated flux.

UEPOCH(200) : double word floating point array into which are stored the epoch times (in seconds of century) from the calibration cards.

UGTBEG(200) : double word floating point array into which are stored all begin times in seconds of century from the good time cards.

UGTEND(200) : double word floating point array into which are stored all end times in seconds of century from the good time cards.

UPRINT : double word floating point variable containing the accumulation time of the first scan to be processed in this run.

UPRT : double word floating point variable containing the accumulation time of the last scan to be processed in this run.

UTMID : double word floating point variable containing the midpoint of accumulation time for a spectrum.

UT1 : double word floating point variable containing the accumulation time of a scan in seconds of century.

UT2 : double word floating point variable containing the accumulation time of the following scan in seconds of century.

UVASOC(30) : double word floating point array containing the most current entries of the time of peak flux (seconds of century).

ZNZ : floating point full word variable into which  $n_0$  is read from the calibration card.

ZNZERO(200) : floating point full word array into which are stored all values of  $n_0$  from the calibration cards.

# VI. Subroutine CALNDR (USOC,IYR,IMO,IDA,IHR,IMN,SEC)

Program Language : Fortran IV, G-Level

Date Written : February 1972

Programmer : S. Croteau

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments : 1) USOC: double precision input floating  
point variable containing seconds of  
century.

2) IYR: output integer full word con-  
taining year.

3) IMO: output integer full word con-  
taining month.

4) IDA: output integer full word con-  
taining day.

5) IHR: output integer full word con-  
taining hour.

6) IMN: output integer full word con-  
taining minutes.

7) SEC: output floating point full word  
containing seconds.

Subprograms Called : None

Description : This subroutine accepts seconds of cen-  
tury as input via the argument list and  
converts this time to calendar time --  
year, month, day, hours, minutes, and  
seconds. The input value in seconds of  
century is a 64-bit floating point word.  
The output variables are 32-bit integers  
except for the seconds which are stored  
in a 32-bit floating point word.



## VII. Subroutine CELEAR (RA,DEC,SATRA,SATDEC,THETA,PHI)

Program Language : Fortran IV, G-Level

Date Written : January 1972

Programmer : Dr. P. Higbie

Installation : University of New Hampshire

Computer : IBM-360-H50

Dummy Arguments : 1) RA: input floating point full word  
variable containing right ascension.

2) DEC: input floating point full word  
variable containing declination.

3) SATRA: input floating point full word  
variable containing satellite right  
ascension.

4) SATDEC: input floating point full word  
variable containing satellite declination.

5) THETA: output floating point full word  
variable containing theta.

6) PHI: output floating point full word  
variable containing phi.

Subprograms Called : SPHREC, ROTATE, RECSPH

Description : This subroutine transforms a vector  
specified in celestial coordinates into  
a vector in earth-centered coordinates.

## VIII. Subroutine CROSS (A,B,AXBN)

Program Language : Fortran IV, G-Level

Date Written : January, 1972

Programmer : P. Higbie

Installation : University of New Hampshire

Computer : IBM-360-H50

Dummy Arguments : 1) A: input floating point array of three values.

2) B: input floating point array of three values.

3) AXBN: output floating point array of three values.

Subprograms Called : None

Description : This subroutine is a vector routine called by subroutine PARM in the calculation of the scan quadrant declination. It does what its name implies.

## IX. Subprogram FEEDER

Program Language : IBM O/S Assembler Language

See Section 1 of this report.

## X. Subroutine FORTIM (IRUNDA,IRUNT)

Program Language : IBM O/S 360 Assembler Language

Date Written : July 1969

Programmer : A. Buck

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments : 1) IRUNDA: output integer full word  
containing the date.  
  
2) IRUNT: output integer full word  
containing the time of day in hundredths  
of a second.

Subprograms Called : None.

Description : Subroutine FORTIM uses the time macro  
to fetch the date and the time from  
the computer clock. The date and the  
time are only as accurate as the cor-  
responding information entered by the  
operator.

Example : FORTIM is called at 2.37 seconds after  
3 minutes after 1 o'clock, A.M. on  
February 10, 1970. When control is  
returned from subroutine FORTIM, IRUNDA  
contains 70041 in binary form and IRUNT  
contains 3780237 in binary form.

## XII Subroutine FPUT (IDSN, ICMD, IRSLT, IOVEC)

Program Language : IBM O/S 360 Assembler Language

Date Written : February 4, 1971

Programmer : A. Buck

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments :

- 1) IDSN: a full word integer input parameter specifying the data set to which activity is directed. The integer stored at IDSN must be greater than zero but less than 21.
- 2) ICMD: a full word input integer variable containing a request type code which specifies the activity to be performed.
  - a) ICMD = 1 implies that a logical record is to be written.
  - b) ICMD = 2 implies the data set is to be ended. Once ended, the data set is unavailable to the calling program through the same data set number.
- 3) IRSLT: a full word output integer variable in which subroutine FPUT can place a return code as follows
  - a) IRSLT = -1 implies request was illegal for one of the reasons below:
    - i. IDSN contained a value less than 1 or greater than 20.
    - ii. ICMD contained a value less than 1 or greater than 2.
    - iii. A write request was made to a data set that had been ended.

iv. An endfile request was made to a data set either before the first write request or after the data set had been ended.

b) IRSLT = 0 implies request was executed normally.

c) IRSLT = 1 implies an I/O error occurred while writing a block. The data set was ended.

4) IOVEC: an input vector whose high order byte is the first byte to be written (when ICMD = 1) in a logical record.

The number of bytes written is specified by the LRECL subparameter on the DD statement describing the data set.

Subprograms Called : None

Description : Subroutine FPUT is a QSAM output subroutine to write logical records from an area defined by the calling program. This routine is specifically designed to allow a Fortran program to create without format control an extensive data set to which the DCB subparameter 'RECFM = FB' applies. For data sets to which 'RECFM = F' or RECFM = U' applies, subroutine FTWRIT may be more easily used.

Subroutine FPUT can manage as many as 20 data sets in the same job step. As many as desired may be active simultaneously. Once the endfile command has been issued to a data set, subroutine FPUT will refuse to perform any further activity for that particular data set. The large number of data sets allows writing as many as 20 files to the same magnetic tape volume as a possible application.

DD Statements required...

If subroutine FPUT is called with IDSN=1 (or 2, ..., or 20) the user must supply a DD Statement having the DD name FPUT01 (or FPUT02, ..., or FPUT20). The DD Statement must include the DCB subparameters 'RECFM', 'LRECL', and 'BLKSIZE' but must omit 'DSORG' and 'MACRF'.

## XII. Subroutine GAINB (CHAN,MONE1,MTWO2)

Program Language : Fortran IV, G-Level  
 Date Written : November 1972  
 Programmer : S. Croteau  
 Installation : University of New Hampshire  
 Computer : IBM-360-H50  
 Dummy Arguments : 1) CHAN: output floating point full word array containing the gain-corrected spectrum.  
 2) MONE1: output integer full word variable containing the channel number of the first element of CHAN.  
 3) MTWO2: output integer full word variable containing the channel number of the last element of the gain-corrected spectrum in CHAN.  
 Subprograms Called : None  
 Description : This subroutine does a gain change normalization in the following manner:

## A. Experimenter tape

where

$n$  = channel number

$N_n$  = counts in channel  $n$

$E_n = C(n + n_0)^2$  = energy of left channel edge

$\Delta E_n = E_{n+1} - E_n = 2C(n + n_0 + \frac{1}{2})$   
 = width of energy window  
 for channel  $n$

$C, n_0$  = determined gain and offset parameters

$\eta_n = N_n / \Delta E_n$  = counts per MeV in channel  $n$ .

## B. Pass-III normalized tape

$$n = > n$$

$$N_n = > N_n$$

$$E_m = K_m^2$$

$$K = \text{fixed gain} = .5 \times 10^{-4}$$

let

$$m_1 = \text{first integer } > \sqrt{\frac{C}{K}} (15 + n_o)$$

$$m_2 = \text{last integer } > \sqrt{\frac{C}{K}} (377 + n_o)$$

### C. Gain Changing

The number of counts in the experimenter tapes is

$$\eta_E = \eta_{15}^{H_{E16}^{E15}} + \eta_{16}^{H_{E17}^{E16}} + \eta_{17}^{H_{E18}^{E17}} + \dots + \eta_{376}^{H_{E377}^{E376}}$$

The number of counts per channel in the Pass-III tape is:

$$M_m = \int_{E_m}^{E_{m+1}} \eta_E dE$$

The inputs to this program are the C value, the standard C(.00005), NZERO(80.2), and the main frame data for either a background or a solar quadrant. The energy is computed for channels 1 to 377 and then the counts/MeV are calculated by dividing the number of counts in a certain channel by the difference in MeV between that channel and the following channel. Then the energy is computed for channels MONE1 to MTWO2. The energies of the two spectra are compared and the program computes channel counts for the new spectrum by multiplying the difference of energies in MeV by counts/MeV. An array of 4 values is used to store all the counts for one particular channel and is summed to give the total counts for that channel. This procedure is followed for all the channels through MTWO2. The new gain-corrected spectrum is then passed to the calling program through CHAN.



## XIII. ENTRY HPARM

Program Language : Fortran IV, G-Level  
Date Written : January 1972  
Programmer : P. Higbie  
Installation : University of New Hampshire  
Computer : IBM-360-H50  
Dummy Arguments : None  
Subprograms Called : FEEDER, VABELT  
Description : ENTRY HPARM is an entry of subroutine PARM. HPARM is used to calculate Van Allen Belt parameters as well as the sun right ascension and declination. The technique is the same as that used in the stand-alone operations Van Allen Predict Program (see Section 6). Both programs used the subroutine VABELT involving data interpolated from World Maps of Flux Contours (NASA SP-3054).

The orbital data used by HPARM is supplied by subprogram FEEDER from the experimenter data tapes. HPARM is called at the beginning of each new file and returns the peak flux, the time and the integrated dose of any Van Allen Belt encounter.

## XIV. ENTRY INTPRM

Program Language : Fortran IV, G-Level  
Date Written : January 1972  
Programmer : P. Higbie  
Installation : University of New Hampshire  
Computer : IBM-360-H50  
Dummy Arguments : None  
Subprograms Called : None  
Description : ENTRY INTPRM is an entry of subroutine PARM. It is called once during program execution prior to any call to PARM or HPARM. It simply serves the purpose of initializing variables and constants to be used in subsequent calls to the other two routines.

## XV. Subroutine PARM

Program Language : Fortran IV, G-Level  
Date Written : January 1972  
Programmer : P. Higbie  
Installation : University of New Hampshire  
Computer : IBM-360-H50  
Dummy Arguments : None  
Subprograms Called : SPHREC, CROSS, RECSPH, REEDER, CELEAR, VCR  
Description : Subroutine PARM calculates the right ascension and the declination for the satellite. The spin and reference direction data is acquired from the header information supplied with each major frame of data. Magnetic coordinate information and the position of the satellite at the time of spectra accumulation is taken from the orbit data supplied by FEEDER from the experimenter data tapes. This information is then used to calculate the reference directions in an earth-centered coordinate system. The subroutine VCR is called to calculate the vertical cutoff rigidity.

## XVI. Subroutine RECSPH (THETA,PHI,V)

Program Language : Fortran IV, G-Level

Date Written : January 1972

Programmer : P. Higbie

Installation : University of New Hampshire

Computer : IBM-360-H50

Dummy Arguments : 1) THETA: output floating point full word variable containing theta.

2) PHI: output floating point full word variable containing phi.

3) V: input floating point array of three elements used to calculate theta and phi.

Subprograms Called : None

Description : This subroutine is called by PARM to calculate THETA and PHI from the 3-element array V.

## XVIII. Subroutine ROTATE (N,AA,V,VP)

Program Language : Fortran IV, G-Level

Date Written : January 1972

Programmer : P. Higbie

Installation : University of New Hampshire

Computer : IBM-360-H50

Dummy Arguments : 1) N: input integer full word variable  
set to 1, 2, or 3 to indicate how the  
axes are to be rotated.

2) AA: input floating point full word  
variable containing the right ascension.

3) V: input floating point full word  
vector of three elements to be rotated.

4) VP: output floating point full word  
vector of three elements.

Subprograms Called : None

Description : This program is called by CELEAR in the  
transformation from celestial coordinates  
to earth-centered coordinates. ROTATE is  
called to establish the new coordinate  
system by rotating the axis.

## XVIII. Subroutine SOCENT (IRY,IMO,IDA,IHR,IMN,SEC,USOC)

Program Language : Fortran IV, G-Level

Date Written : February 1972

Programmer : S. Croteau

Installation : University of New Hampshire

Computer : IBM-360-H50

Dummy Arguments : 1) IYR: input integer full word variable  
containing year.

2) IMO: input integer full word variable  
containing month.

3) IDA: input integer full word variable  
containing day.

4) IHR: input integer full word variable  
containing hour.

5) IMIN: input integer full word variable  
containing minutes.

6) SEC: input floating point full word  
variable containing seconds.

7) USOC: output double word floating  
point variable containing seconds of  
century.

Subprograms Called : None

Description : This subroutine converts calendar time  
to seconds of century.

## XIX. Subroutine SPHREC (THETA,PHI,V)

Program Language : Fortran IV, G-Level

Date Written : January 1972

Programmer : P. Higbie

Installation : University of New Hampshire

Computer : IBM-360-H50

Dummy Arguments : 1) THETA: input floating point full word  
variable containing theta.  
2) PHI: input floating point full word  
variable containing phi.  
3) V: output floating point full word  
array of 3 elements.

Subprograms Called : None

Description : This subroutine is called by CELEAR and  
by PARM. It accepts as input THETA and  
PHI and returns the 3-element array V for  
values of theta from -10.0 to 10.0.

## XX. Subroutine VABELT (GLON, GLAT, ALT, FLUX)

Program Language : Fortran IV, G-Level

Date Written : December 1971

Programmer : P. Higbie

Installation : University of New Hampshire

Computer : IBM-360-H50

Dummy Arguments : 1) GLON: input floating point full word variable containing the longitude.

2) GLAT: input floating point full word variable containing the latitude.

3) ALT: input floating point full word variable containing the altitude.

4) FLUX: output floating point full word variable containing the flux.

Subprograms Called : None

Description : Data for the VABELT program was taken by interpolating at 15° intervals the data shown in the document, "World Maps of Constant B, L, and Flux Contours," (NASA SP-3054). The data is for integral proton fluxes for E>5 MeV.

Linear interpolation on the grid defined by the array FLXGRD is performed according to techniques given in McCormick and Salvadori, Numerical Methods in Fortran, page 118. The argument list consists of the longitude of the satellite in degrees, the latitude of the satellite in degrees, and the altitude of the satellite in kilometers. The program returns the proton flux at the given coordinates (counts/cm<sup>2</sup>sec).



## XXI. Subroutine VCR (GLON, GLAT, PC)

Program Language : Fortran IV, G-Level  
Date Written : December 1971  
Programmer : P. Higbie  
Installation : University of New Hampshire  
Computer : IBM-360-H50  
Dummy Arguments : 1) GLON: input floating point full word  
variable containing the longitude.  
2) GLAT: input floating point full word  
variable containing the latitude.  
3) PC: output floating point full word  
variable containing the rigidity.  
Subprograms Called : None  
Description : Subroutine VCR performs an interpolation  
on the map of cutoff rigidities given by  
Shea et al. The latitude and longitude  
must be supplied as input parameters; the  
vertical cutoff rigidity (PC) is returned.

## XXII. Description of Input

### A. Control card format

This card must be the first card read. The format is as follows

[illegible]

NTAPES

Number of tapes to be processed

MFDOP

```
0 = Main frame data not processed
1 = All main frame data processed
2 = PHA channel counts ignored
```

IATTOP

0 = Omission of attitude data  
1 = Attitude data values processed

IDSOP

0 = Omission of digital subcom data  
1 = Digital subcom data processed

IASOP

0 = Omission of wheel analog subcom  
words 3 - 10  
1 = All 10 wheel analog words to be  
processed

TORBOP

```

0 = No orbit data needed
1 = Regular orbit data items are
desired
2 = All orbit data items required

```

IF1  
IF2  
IF3

IF1 = 1 IF2 = 10 IF3 = 15  
These input values are used in the determination of the presence or absence of bit errors in the gamma spectrum. They were included in the control card so as to simplify a re-definition of the requirements as to what constitutes a bit error. As it turned out, the original definition and requirements proved to be quite adequate.

[illegible]

YEAR

MONTH

DAY

HOUR

MIN.

SEC

MAX FLUX

INTEGRATED FLUX

ID

The calendar time representing the accumulation time of the maximum flux.

The value corresponding to the maximum flux for a given orbit.

The integrated flux for a given orbit.

= '400' for Van Allen cards



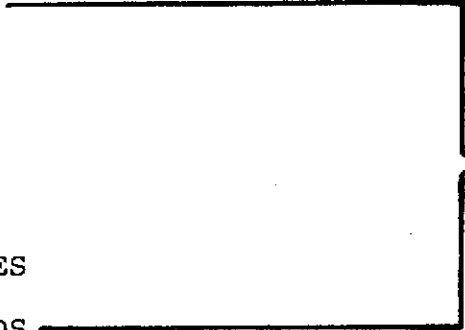
BO/BK

Indicates whether to stop processing at the background scan or after processing both background and solar.

ID

= '100' for good times cards.



YEAR		
MONTH		
DAY		
HOURS		
MINUTES		
SECONDS		
		The epoch time
q		The value of $C_o$
r		The value of $\frac{dC}{dt}$
$\bar{n}_o$		The average $n_o$ (80.2)
QUAL		The quality indicator
		'1' = good quality '2' = questionable quality '3' = poor quality
s		The second derivative. Not equal to zero only if there were more than three calibration runs for the same epoch time - it is calculated using three equations.
ID		= '300' for calibration card type 1. = '301' for calibration card type 2.



### E. Experimenter Data Tape

Pass-III reads the experimenter tapes using Subprogram FEEDER. For a description of the tape format, see Section 2 XXIII B of this report.

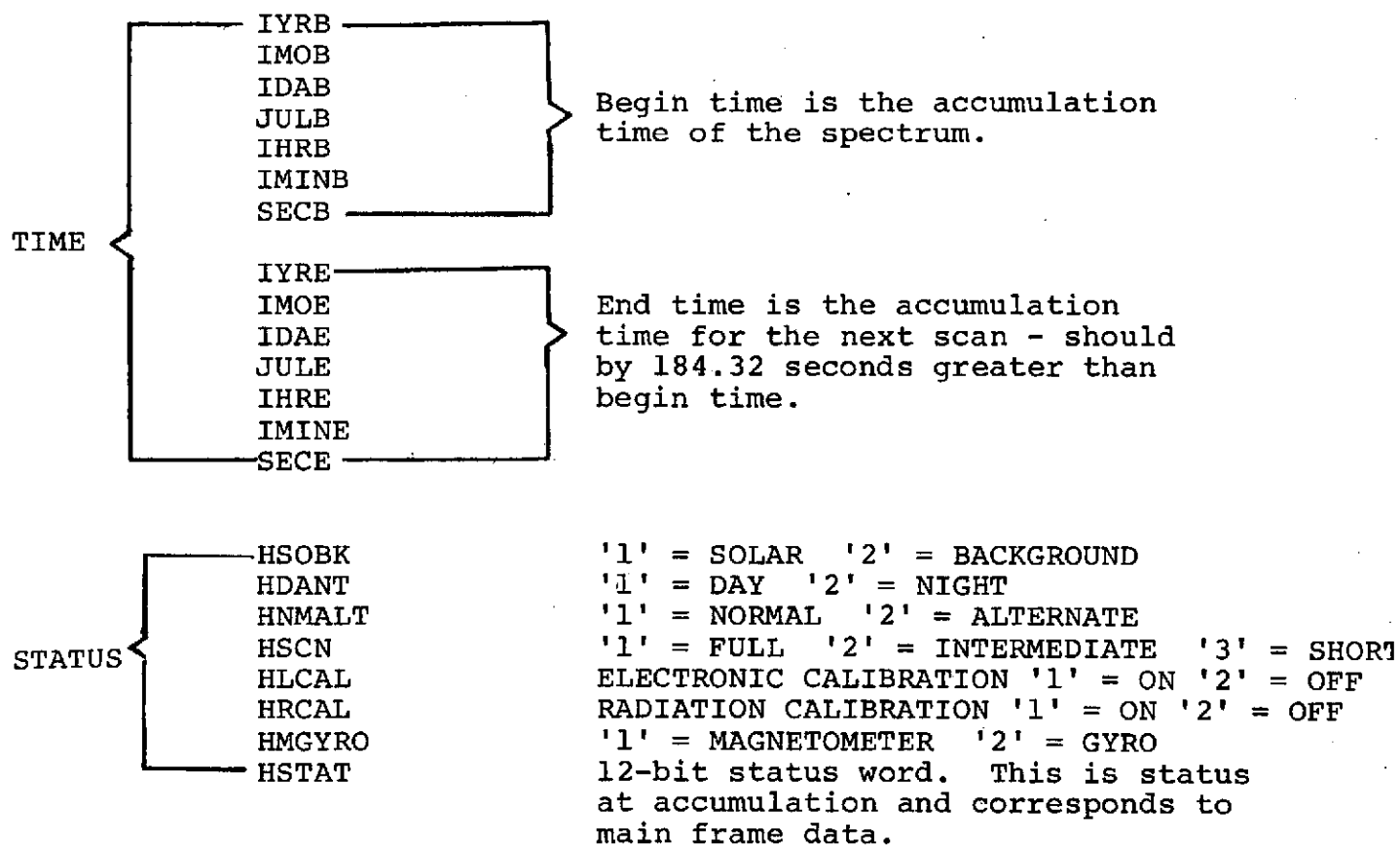
## XXIII. Description of Output

The output of this program is a 9-track parameter tape with a logical record length of 3540 and a block size of 3540 8-bit bytes.

The first letter of each variable described below indicates its type as follows:

A-G: 32-bit floating point word.  
 H : 16-bit integer.  
 I-N: 32-bit integer.  
 O-T: 32-bit floating point word.  
 U : 64-bit character.  
 V-Z: 32-bit floating point word.

The type format is as follows:



	RIGID	Vertical cutoff rigidity at time of midpoint.
POSITION INFORMATION	BGAUS ERMSIL	B and L magnetic coordinates.
	XLAT XLONG XALT	Latitude Longitude Altitude
VARIABLE OBJECTS	RASAT DECSAT RASUN DECSUN	Satellite right ascension Satellite declination Sun right ascension Sun declination
CELESTIAL COORDINATES	RASQD DECSQD RASPIN DECSPN	If solar: Right ascension sun quadrant If background: Right ascension background quadrant If solar: Declination sun quadrant If background: Declination background quadrant Right ascension spin axis Declination spin axis
EARTH COORDINATES	THESUN PHISUN THETAS PHIS SESANG AHORIZ	If solar: Theta sun quadrant If background: Theta background quadrant If solar: Phi sun quadrant If background: Phi background quadrant Theta spin axis Phi spin axis Satellite-earth-sun angle = RASAT - RASUN Horizon angle $= \tan^{-1} \left( \frac{\text{radius earth}}{R\oplus + \text{Alt}} \right) * \frac{180}{\pi}$
	RXRA(1)	BACKGROUND = Total X-ray counts in channel 1 $= \frac{\Sigma I \ 29}{\Sigma II - 24}$
	RXRA(2)	SOLAR = Total X-ray counts in channel 1 $= \frac{\Sigma I \ 23}{\Sigma II - 22}$

VAN ALLEN PARAMETERS	IYRVPK(14)	Time of Van Allen peaks.
	IMOVPK(14)	
	IDAVPK(14)	
	JULVPK(14)	
	IHRVPK(14)	
	IMNVPK(14)	
	SECVPK(14)	
	FLUX(14)	Flux
FLXINT(14)	Integrated flux	
UTAPE	Tape volume serial number	
IFILE	Decomm run number	
ICYRDA	Run date: year and day of year	
ICRHRM	Run date: hours and minutes	
MONE1	Output from gain shift - 1st channel	
MTWO2	Output from gain shift - last channel	
TLIVE	$TCLOCK = \text{Clock time} * 1/390.625 * 8$ $TLIVE = TCLOCK - TDEAD$	
TDEAD	Deadtime = $DEADTIME * .001515$ (+ 24.90 if fold over present)	
CHAN(700)	Gain-shifted spectrum	
HBITER	Bit error indicator = '0' if no bit errors present = '1' if at least 1 bit error found	
HFOLD	Indicates dead time fold over = '0' if no fold over present = '1' if rigidity $\leq 60$ and dead time $< 10.0$ . When $HFOLD = 1$ , the dead time has been adjusted by addition of 24.90 to the original value.	

HRATE  $= \frac{A + B}{D + E}$  where

A = ch. 151 to 155

B = ch. 161 to 165

D = ch. 144 to 148

E = ch. 168 to 172

HAVRTE  $= \frac{\text{HRATE}}{\text{HDIV}}$  (No. of entries)

There is one average for solar quadrants and one average for background quadrants.

HDIV The number of scans for which rates are included to calculate HAVRTE. Starts over after every time gap and after every calibration.

HFLGS Number of flagged main frame data words.

HQUAL Quality indicator - indicates quality of gain shift.

HATT = '0' if attitude data was available  
= '1' if no attitude data was available

HORB = '0' if orbit data was available  
= '1' if no orbit data was available

HPRM = '0' if SUNRA and SUNDEC are valid  
= '1' if SUNRA and SUNDEC are not valid

HIO = Number of I/O error major frames in section

HEXTRA(9) Extra words to be used at a later date.

Sample Output  
PASS-III

NUMBER OF CARDS WITH INVALID ID = 0

TAPE IDENTIFICATION DECOMM RUN NUMBER = 158 VOLUME=SER=EXD314

FIRST GOOD TIME YEAR=72 MONTH=12 DAY= 6 HOUR= 1 MINUTES=18 SECONDS=46.8

RUN DATE YR / DAY OF YR 73249 HRS / MIN 2008

TAPE IDENTIFICATION RUN NUMBER = 158 VOLUME=SER=EXD315

TAPE IDENTIFICATION RUN NUMBER = 159 VOLUME=SER=EXD292

TAPE IDENTIFICATION RUN NUMBER = 159 VOLUME=SER=EXD293

TAPE IDENTIFICATION RUN NUMBER = 160 VOLUME=SER=EXD316

TAPE IDENTIFICATION RUN NUMBER = 161 VOLUME=SER=EXD294

TAPE IDENTIFICATION RUN NUMBER = 161 VOLUME=SER=EXD295

TAPE IDENTIFICATION RUN NUMBER = 162 VOLUME=SER=EXD306

LAST GOOD TIME YEAR=72 MONTH=12 DAY=17 HOUR= 0 MINUTES=58 SECONDS=22.8

NUMBER RECORDS WRITTEN = 5370

## Section 6

### Van Allen Belt Predict Program.

	<u>Page</u>
I. Introduction	6-1
II. Macro Systems Flowchart	6-2
III. Job Stream Schematic	6-3
IV. Job Control Language	6-4
V. Disk Preparation Program	6-5
VI. VAPREDICT Main Program	6-7
VII. Subroutine HEIGHT (GLAT,R,ALT)	6-8
VIII. Subroutine VABELT (GLON,GLAT,ALT,FLUX)	6-9
IX. Description of Input	6-10
X. Description of Output	6-19
XI. Sample Output Listing	6-20

## I. Introduction

This program is a stand-alone program which computes certain parameters for the satellite's passage through the Van Allen Belt. The basic subroutines called are HEIGHT and VABELT which are also used in the primary data analysis.

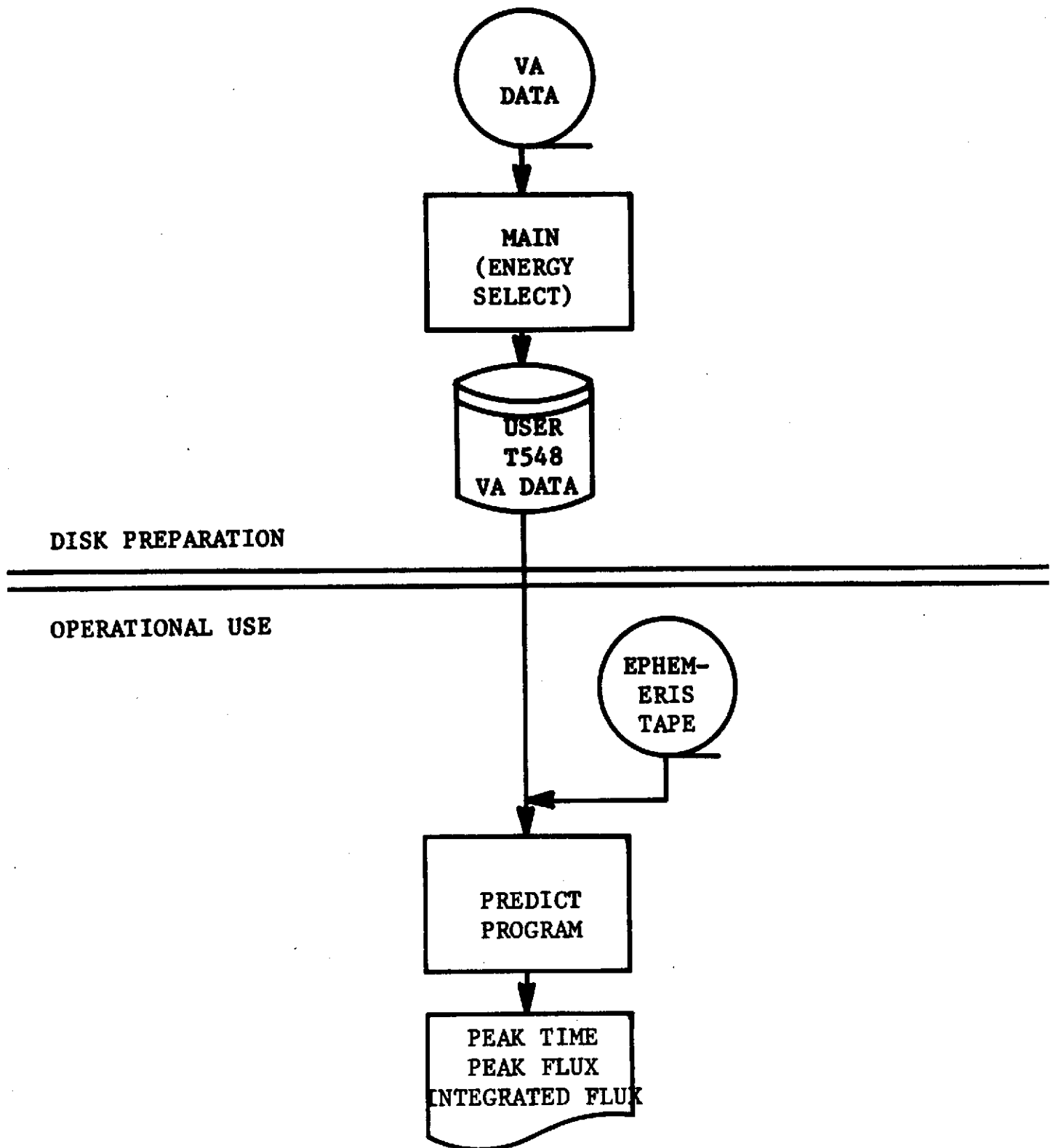
A data set is prepared from data supplied by NASA (E.G. Stassinopolous) in a preparatory step and stored on disk. Normal operational use requires an Ephemeris data tape supplied by NASA (J.B. Joyce) and the data set stored on disk.

The predict program computes for a given period of time the peak flux encountered, the time of the peak flux, the integral flux during the orbit, the times where the satellite encountered and then left regions with fluxes  $>10^3$  photons/cm<sup>2</sup>sec with the corresponding positional information.

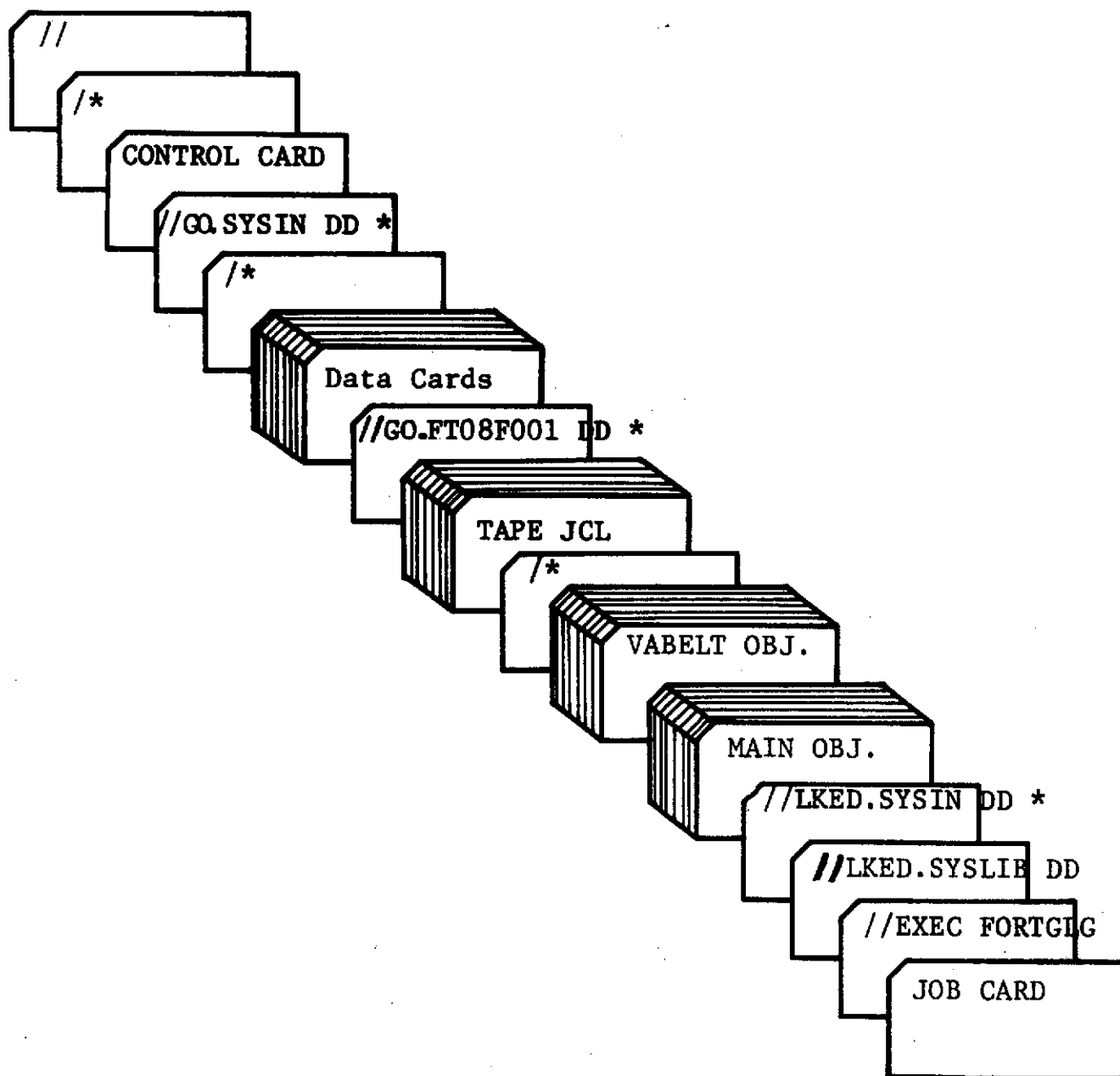
Estimated uncertainties are less than one minute for the orbital elements, three weeks from epoch. At the three-sigma confidence level the VA data is off by a factor of 2 at worst.



## II. Macro System Flowchart (Van Allen Predict)



## III. Job Stream Schematic (Van Allen Predict)



## IV. JOB CONTROL LANGUAGE - VAPREDICT

```
//          T548),'VAPREDICT,INKIE G.',CLASS=F,TIME=10,REGION=140K
//STEP1 EXEC FORTGLG
//LKED.SYSLIB DD DSN=USER.T548.SUBLIB,DISP=SHR
//          DD DSN=USER.T317.SUBLIB,DISP=SHR
//          DD DSN=SYS1.FORTLIB,DISP=SHR
//          DD DSN=USER.SSPLIB,DISP=SHR
//LKED.SYSIN DD *
```

\*\*\* VAPREDICT AND VABELT OBJECT DECKS \*\*\*

```
/*
//GO.FT10F001 DD DISP=OLD,DSN=USER.T548.VADATA
//GO.FT09F001 DD UNIT=2400,DCB=(RECFM=VS,BLKSIZE=2808),LABEL=(1,BLP),
//          VOLUME=SER=EFM00A
//GO.SYSUDUMP DD SYSOUT=A
//GO.FT08F001 DD *
3.          2.          50.
270.        51.        300.
60.         270.        9.
/*
//GO.SYSIN DD *
72
/*
//
```

## V. Disk Preparation Program

Program Language : Fortran IV, G-Level  
Date Written : September 1971  
Programmer : P. Higbie  
Installation : University of New Hampshire  
Computer : IBM-360-H50  
Subprograms Called : None  
Description : This program is used to generate the array FLXGRD (flux grid) and to put it out on disk on the data set named USER.T548. VADATA. This grid remains on disk throughout the use of the Van Allen Predict program.

The data set is prepared from data supplied by NASA (E. G. Stassinopolous). Energy selection may be accompanied by changing the variable equated to FLXGRD. The listing has F15 for this variable which corresponds to a flux map of protons greater than 15 MeV in energy. Corresponding to the various FN's are the energies N=3,5,15,30,50,100 MeV.

A listing of the program is given for reference (Figure V.1).

N IV G LEVEL 19

MAIN

DATE = 72095

13/57/01

```
DIMENSION ALT (7),FLXGRD(51,49,7)
DO 10 IALT=1,7
READ(9,300)ALT(IALT)
300  FORMAT(2F6.1,F8.5,F6.2,6E12.4)
DO 10 ILAT=1,49
DO 10 ILON=1,51
JLAT=50-ILAT
READ(9,300)GLAT,GLON,B,XL,F3,F5,F15,F30,F50,F100
10  FLXGRD(ILON,JLAT,IALT)=F15
WRITE(10)FLXGRD
END FILE 10
END
```

```
//GO.FT10F001 DD UNIT=SYSDA, DISP=(NEW,CATLG),
               DSN=USER,TS48,VADATA,VOL=SER=USER01
```

Figure V.1. Disk Preparation Program

## VI. VAPREDICT Main Program

Program Language : Fortran IV, G-Level

Date Written : September 1971

Programmer : P. Higbie

Installation : University of New Hampshire

Computer : IBM-360-H50

Subprograms Called : HEIGHT, VABELT

Description : This program reads the ephemeris data tape, calculates the longitude and the latitude, calls HEIGHT to calculate the altitude, and calls VABELT to calculate the flux. The program computes for a given period of time the peak flux encountered, the time of the peak flux, the integral flux during the orbit, the times when the satellite encountered and the left regions with fluxes  $>10^3$  photons/cm<sup>2</sup> sec with the corresponding positional information. A listing is then generated containing all of the information mentioned above.

## VII. Subroutine HEIGHT (GLAT,R,ALT)

Program Language : Fortran IV, G-Level  
Date Written : September 1971  
Programmer : P. Higbie  
Installation : University of New Hampshire  
Computer : IBM-360-H50  
Dummy Arguments : 1) GLAT: input floating point full word  
variable containing latitude.  
2) R: input floating point full word  
variable related to time.  
3) ALT: output floating point full word  
variable containing altitude.  
Subprograms Called : None  
Description : This subroutine is responsible for the  
calculation of the altitude of the  
satellite at a certain latitude.

## VIII. Subroutine VABELT (GLON, GLAT, ALT, FLUX)

Program Language : Fortran IV, G-Level  
 Date Written : December 1971  
 Programmer : P. Higbie  
 Installation : University of New Hampshire  
 Computer : IBM-360-H50  
 Dummy Arguments : 1) GLON: input floating point full word variable containing the longitude.  
                   2) GLAT: input floating point full word variable containing the latitude.  
                   3) ALT: input floating point full word variable containing the altitude.  
                   4) FLUX: output floating point full word variable containing the flux.  
 Subprograms Called : None  
 Description : Data for the VABELT program was taken by interpolating at  $15^\circ$  intervals the data shown in the document, "World Maps of Constant B, L, and Flux Contours," (NASA SP-3054). The data is for integral proton fluxes for  $E > 5$  MeV.  
               Linear interpolation on the grid defined by the array FLXGRD is performed according to techniques given in McCormick and Salvadori, Numerical Methods in Fortran, page 118. The argument list consists of the longitude of the satellite in degrees, the latitude of the satellite in degrees, and the altitude of the satellite in kilometers. The program returns the proton flux at the given coordinates (counts/cm<sup>2</sup>sec).



# IX. Description of Input Ephemeris Data Tape

This input tape is a 9-track tape (RECFM = VS, BLKSIZE = 2808)

ENTRY DESCRIPTION      RECORD 1	INTERNAL			EPHEM OUTPUT	
	BYTES	FORMAT	UNITS	FORMAT	UNITS
(1) Type of tape ID (The word "EPHEM" in EBCDIC)	8	LF	-	EBCDIC	-
(2) Satellite number (7 digits)	4	LI	-	LF	-
(3) UTC, A.1 indicator; = 1, A.1 = 2, UTC	4	LI	-	LF	-
(Items 4, 5, and 6 given Start time of the Ephemeris)					
(4) Date					Year, Month, Day
(5) Day count of year	8	LF	DUT	LF	Days from Jan. 0
(6) Seconds of day				LF	Seconds
(Items 7, 8, and 9 give End time of the Ephemeris)					
(7) Date				LF	Year, Month, Day
(8) Day count of year	8	LF	DUT	LF	Days from Jan. 0
(9) Seconds of day				LF	Seconds

ENTRY DESCRIPTION      RECORD 1	INTERNAL			EPHEM OUTPUT	
	BYTES	FORMAT	UNITS	FORMAT	UNITS
(10) $\Delta t$ , interval between ephemeris points	8	LF	DUT	LF	Seconds
(11) -(26) Run ID taken from MIR comment cards	8	LF	-	EBCDIC	(Up to 128 characters)
(27) Reference day for DODS (September 15, 1957)	8	LF	Calendar	LF	Year, Month, Day
(28) Type of Coordinate System Used for Output = 1, True Equator and Equinox plus Equation of Equinoxes, = 2, Mean Equator and Mean Equinox = 3, True Equator and True Equinox	4	LI	-	EBCDIC and LI	(1st 4 Bytes EBCDIC, 2nd 4 Bytes Indicator value): "True" 1 "Mean" 2 "Inter" 3
(29) Orbit Theory Used: = 1, Cowell = 2, Brouwer	4	LI	-	EBCDIC	(8 Bytes Saying "Cowell" or "Brouwer")
(30) Solar Radiation Constant	4	SF	$\frac{gm}{DUT^2}$	LF	$\frac{gm}{cm \ sec^2}$
(31) Solar Flux Number	4	SF	-	LF	-
(32) $C_D$ , Coefficient of Drag	4	SF	-	LF	-

ENTRY DESCRIPTION      RECORD 1	INTERNAL			EPIHEM OUTPUT	
	BYTES	FORMAT	UNITS	FORMAT	UNITS
(33) $C_R$ , Shape	8	LF	-	LF	-
(34) Atmosphere Model ID	8	LF	-	EBCDIC	(8 bytes saying "HAR-PRIE")
(35) Cross sectional area of satellite	8	LF	DUL <sup>2</sup>	LF	cm <sup>2</sup>
(36) Mass of satellite	8	LF	gm	LF	gm
(37) Zonal and tessera harmonics indicate	4	LI	-	LF = 0, used = 1, not used	-
(38) Brouwer					
(39) Lunar gravitation perturbation					
(40) Solar radiation perturbation					
(41) Solar gravitation perturbation					

ENTRY DESCRIPTION	RECORD 1	INTERNAL			EPHEM OUTPUT	
		BYTES	FORMAT	UNITS	FORMAT	UNITS
(42)	Drag perturbation indicator	4	LI	-	LF	-
(43)	Elements epoch, $t_0$	8	LF	DUT	LF	DUT
(44)	Year				LF	Year
(45)	Month				LF	Month
(46)	Day				LF	Day
(47)	Hour	8	LF	DUT	LF	Hour
(48)	Minute				LF	Minute
(49)	Seconds X 1000				LF	Seconds
(Items 50 through 55 are Brouwer mean if item 29 = 'Brouwer', classical Keplerian if item 29 = 'Cowell')						
(50)	Semi-major axis at $t_0$	8	LF	DUL	LF	km.
(51)	Eccentricity at $t_0$	8	LF	-	LF	-
(52)	Inclination at $t_0$	8	LF	rad	LF	rad

ENTRY DESCRIPTION      RECORD 1	INTERNAL			EPHEM OUTPUT	
	BYTES	FORMAT	UNITS	FORMAT	UNITS
(53) Argument of perigee at $t_0$	8	LF	rad	LF	rad
(54) Node at $t_0$	8	LF	rad	LF	rad
(55) Mean anomaly at $t_0$	8	LF	rad	LF	rad
(Items 56 through 71 are osculating values evaluated at $t_0$ )					
(56) True Anomaly	8	LF	rad	LF	rad
(57) Arg. of lat. -(53) +(56)	8	LF	rad	LF	rad
(58) Flight path angle	8	LF	rad	LF	rad
(59) Eccentric Anomaly	8	LF	rad	LF	rad
(60) Period	8	LF	DUT	LF	DUT
(61) Perigee height	8	LF	DUL	LF	km
(62) Apogee height	8	LF	DUL	LF	km
(63) Mean motion	8	LF	rad/DUT	LF	rad/DUT

ENTRY DESCRIPTION	RECORD 1	INTERNAL			EPHEM OUTPUT	
		BYTES	FORMAT	UNITS	FORMAT	UNITS
(64)	Argument of perigee secular rate of change	8	LF	rad/DUT	LF	rad/DUT
(65)	Node secular rate of change	8	LF	rad/DUT	LF	rad/DUT
(66)	X Epoch	8	LF	DUL	LF	DUL
(67)	Y Position					
(68)	Z Vector					
(69)	X Epoch	8	LF	DUL/DUT	LF	DUL/DUT
(70)	Y Velocity					
(71)	Z Vector					
(72)	-(91) Times of Brouwer drag coefficients; $t_{p,0}, t_{p,1}, \dots, t_{p,19}$	8	LF	DUT	LF	DUT
(92)	-(111) Brouwer first order drag coefficients; $N_{2,0}, \dots$ $N_{2,19}$ or Cowell drag coefficients	8	LF	$\frac{\text{rad}}{\text{DUT}^2}$ or units of $\rho_i$		$\frac{\text{rad}}{\text{DUT}^2}$ or internal units of $\rho_i$

ENTRY DESCRIPTION      RECORD 1	INTERNAL			EPHEM OUTPUT	
	BYTES	FORMAT	UNITS	FORMAT	UNITS
(112)-(131) Brouwer second order drag coefficients; N <sub>3,0</sub> , N <sub>3,1</sub> , ... N <sub>3,19</sub>	8	LF	$\frac{\text{rad}}{\text{DUT}^3}$	LF	$\frac{\text{rad}}{\text{DUT}^3}$
(132) Number of non-zero times in items (72)-(91)	4	LI	-	LF	-
(133)-(192) Unknowns associated with epoch elements if their origin was previous DC or PCE	8	LF	-	-	(60 8-Byte EBCDIC words; one for each unknown used)
(193) Start time of ephemeris	8	LF	DUT	LF	DUT
(194) End time of ephemeris	8	LF	DUT	LF	DUT
(195) Δt, interval between ephemeris points	8	LF	DUT	LF	DUT
(196) LPN indicator = 1, include precession and nutation = 2, include precession only = 3, neglect precession and nutation	4	LI	-	LF	-
(197) Greenwich hour angle at epoch	8	LF	Rad	LF	Rad
(198)-(293) - not used (zeros)	8	LF	-	LF	-

ENTRY DESCRIPTION      RECORD 1	INTERNAL			EPHEM OUTPUT	
	BYTES	FORMAT	UNITS	FORMAT	UNITS
(1) Date of 1st ephem. point	8	LF	DUT	LF	Year, Mo., time, t, of Day                      first ephemeris point
(2) Day count of year				LF	Days from Jan. 0
(3) Seconds of day				LF	Seconds
(4) Interval between data items, $\Delta t$	8	LF	DUT	LF	Seconds
(5) X                      position	8				
(6) Y                      vector	8	LF	DUL	LF	DUL
(7) Z	8				
(8) X                      velocity	8				
(9) Y                      vector	8	LF	$\frac{DUL}{DUT}$	LF	$\frac{DUL}{DUT}$
(10) Z	8				
(11)-(304) KK-1 other ephemeris position and velocity vector sets and end-of-data indicators if KK<50					



ENTRY DESCRIPTION      RECORD 1	INTERNAL			EPHEM OUTPUT	
	BYTES	FORMAT	UNITS	FORMAT	UNITS
(305)    t	8	LF	DUT	LF	DUT
(306)     Δt	8	LF	DUT	LF	DUT
(307)-(350)    not used (zeros)	8 each				
(1)-(10) SENTINEL = +.9999999999999999+16	8	LF	-	LF	-
(11)-(350)    not used	8 each	-	-	-	-

## X. Description of Output

The output of the Van Allen Predict program is a listing consisting of the following fields:

- A. Year
- B. Julian Day
- C. Month
- D. Day of month
- E. Hour/minutes
- F. Peak flux in  $\text{cm}^{-2}\text{sec}^{-1}$
- G. Integral flux (dose) in  $\text{cm}^{-2}$
- H. Time and position in which the flux becomes  $> 10^3\text{cm}^{-2}\text{sec}^{-1}$ . This includes the start time in day of year, hour/minutes, the longitude, latitude, and altitude.
- I. Time and position in which the flux fell back below  $10^3\text{cm}^{-2}\text{sec}^{-1}$ . This includes end time in day of year, hour/minutes, the longitude, latitude, and altitude.

Please note that the output identifies times where the satellite encountered flux  $> 10^3\text{cm}^{-2}\text{sec}^{-1}$  as well as times where VA passage occurred but with flux  $< 10^3\text{cm}^{-2}\text{sec}^{-1}$ . Time gaps may be observed where there were no VA encounters.



## Section 7

### University of New Hampshire Computation Center PLOT Routine

	<u>Page</u>
I. Introduction	7-1
II. PLOT Program Descriptions	7-2
III. PLOT Programming Procedure	7-7
IV. Annotation (CHAR) Subroutine	7-8
V. CHAR Programming Procedures	7-9
VI. The Care and Feeding of the Plotter Programs	7-10

## I. Introduction

The IBM-360 PLOT and CHAR routines enable the programmer who uses graphic output to take advantage of the speed and facilities of the 360 for calculations and use the IBM-1620/1627 for plotting only. PLOT and CHAR on the IBM-360 produces the plotter commands (i.e., number strings) and puts them into a 7-track tape which can later be read and plotted by the IBM-1620/1627.

An \* appears in the right-hand margin of this paper by those sections and remarks which are unique to the 360 PLOT and CHAR routines. The programmer who is already familiar with the 1620 PLOT and CHAR routines need only read those sections.

## II. PLOT Program Descriptions

The following definitions apply to the parameters used in a call to PLOT. All X and Y values are to be expressed in inches. The subroutine processes floating point values in the short format.

- A. IC = control integer which must have one of the following values: 1, 101, 201, 0, 9, 10, 19, 20, 29, 30, 39, 40, 49, 50, 59, 90, 99, 7, 100. See "Use of Control Integer" under the specific programming procedure in question.
- B. XMIN equal to or less than minimum X value to be plotted.
- C. XMAX equal to or greater than maximum X value to be plotted.
- D. XL = the required physical length of the plot in the X direction.
- E. XD = X increment to be indicated on plot outline =  $(XMAX - XMIN)/INC$ , where INC equals the line of increments desired.
- F. YMIN equal to or less than minimum Y value to be plotted.
- G. YMAX equal to or greater than maximum Y value to be plotted.
- H. YL = the required physical length of the plot in Y direction.
- I. YD = Y increment to be indicated on plot outline =  $(YMAX - YMIN)/INC$ , where INC equals the number of increments desired.

Choice of variable names for the above information is left to the discretion of the programmer, except for the restrictions that the control integer must be a fixed point variable or constant and the rest be floating point variables (with or without subscripts), constants, or expressions. Array names must not be used.

In brief, PLOT has three sections:

1. Framing and scaling
2. Point-to-point plotting
3. Reinitialization

a. Framing and Scaling Section

In this section, scaling constants are calculated and a plot outline or grid is drawn governed by the control integer IC. These scaling constants are then used to scale all X and Y values in the point-to-point plot section. Therefore, it is imperative that a call to PLOT with IC = 1, 101, 201 be made before any actual plotting is done.

After the frame is drawn the pen is placed in the up status.

Use of the Control Integer (IC)

One of the following three control integers must be used in this section:

IC = 1

IC = 101

IC = 201

When IC = 1 is used, scale constants are calculated and a plotting area outline is drawn. When IC = 101 is used, scale constants are calculated and a grid over the plotting area is drawn. When IC = 201 is used, scale constants are calculated, but neither a plotting area outline nor grid is drawn. (For efficient programming, set XD equal to XMAX - XMIN, and YD equal to YMAX - YMIN, when IC = 201 is used.) The latter control integer is used when a reference frame is already available or when one is not required.

b. Point-to-Point Section

This section of PLOT scales a given point (X,Y) to correspond to the frame established.

Use of Control Integer (IC)

One of the following fourteen control integers must be used in this section:



NOTE: The pen always moves to the point to be plotted in the status in which it was left (up or down) from the previous call.

IC = 0	IC = 10	IC = 30	IC = 50
IC = 9	IC = 19	IC = 39	IC = 59
IC = 90	IC = 20	IC = 40	

- IC = 00. When IC = 0 is used, the point (X,Y) is scaled, the pen moves to the resultant coordinate (X plot, Y plot), and a plus sign (+) is drawn through the point. The pen remains down after the plus sign is drawn
- IC = 9. When IC = 9 is used, the point (X,Y) is scaled, the pen moves to the resultant coordinate (X plot, Y plot), and a plus sign (+) is drawn through the point.
- IC = 90. When IC = 90 is used, the point (X,Y) is scaled, and the pen moves to the resultant coordinate (X plot, Y plot). Then pen is commanded down when the point to be plotted is reached.
- IC = 99. When IC = 99 is used, the pen is commanded up. No other action takes place.
- IC = 10. The same as IC = 0 except that a square (■) is drawn through the point.
- IC = 19. The same as IC = 9 except that a square (■) is drawn through the point.
- IC = 20. The same as IC = 0 except that an octagon (⊠) is drawn through the point.
- IC = 29. The same as IC = 9 except that an octagon (⊠) is drawn through the point.
- IC = 30. The same as IC = 0 except that a diamond (◆) is drawn through the point.
- IC = 39. The same as IC = 9 except that a diamond (◆) is drawn through the point.
- IC = 40. The same as IC = 0 except that a triangle (▲) is drawn through the point.
- IC = 49. The same as IC = 9 except that a triangle (▲) is drawn through the point.

- IC = 50.      The same as IC = 0 except that a flag (P) is drawn through the point.
- IC = 59.      The same as IC = 9 except that a flag (P) is drawn through the point.

NOTE: If the pen up or down status is altered manually or by some routine other than PLOT (for example, the CHAR routine) IC = 99 should be executed prior to the resumption of point-to-point plotting. The reason for this lies in the fact that PLOT is designed to keep track of the pen up or down status so that the pen does not have to be commanded to assume its present status. (One hundred ms are required for each pen up or down command.) By executing an IC = 99 call, as just described, pen up status is established physically and within the program.

c. Reinitialization Section

IC = 7  
IC = 100

After all points have been plotted within a frame, IC = 7 may be called if a new frame is to be drawn. Calling IC = 7 causes the pen to move in the up status to a point 2.5 inches beyond the frame.

After all plotting is complete for the entire program, IC = 100 should be called so that the PLOT program can put the final record onto tape. If IC = 100 is not called, the last part of the plot will be lost.

### III. PLOT Programming Procedure

Prior to calling PLOT, the programmer must determine and place in storage the XMIN, XMAX, XL, SD, YMIN, YMAX, YL, YD, and IC information as previously defined.

The actual names of the variables are unimportant as long as IC is a fixed-point variable or constant, and the rest of the variables are floating-point variables, constants, or expression (except that array names must not be used).

Because of the make-up of the plot routine, the first time it is entered the following statement must be used:

```
CALL PLOT (IC,XMIN,XMAX,XL,XD,YMIN,YMAX,YL,YD)
```

where IC must equal 1, 101, or 201.

After the initial statement has been executed, the subroutine can be entered at any time by one of the following two statements:

```
CALL PLOT (IC,X,Y)
```

```
CALL PLOT (IC)
```

where IC in the first statement must be 0, 9, 10, 19, 20, 29, 30, 39, 40, 49, 50, 59, or 90 and X,Y are the coordinates of the point to be plotted. IC in the second statement must be 7, 99, or 100.

#### IV. Annotation (CHAR) Subroutine

CHAR allows a FORTRAN program to plot letters, numbers, and all FORTRAN special characters from information supplied by FORMAT statements.

##### a. Program Description

The position of the pen, at the time CHAR is called, is at the lower left-hand corner of the character to be drawn. Because the information to be plotted is obtained from FORMAT statements furnished by the programmer, actual variable values can be plotted at object time.

Each character that is plotted is represented in the subroutine by a string of digits. These digits, when transmitted to the plotter, cause the proper pen and paper motion required to draw the desired character. When all of the specified characters are drawn, the pen is returned to the starting position with the pen in the up status.

A maximum of 100 characters (including blanks) can be drawn on any one call to CHAR.

C-4

## V. CHAR Programming Procedures

The CHAR subroutine is accessed by the following three statements:

```
n  FORMAT (S1,S3,S3,...SN,100X)

    WRITE (4,n) VAR1,VAR2,VAR3,...VARN

    CALL CHAR (SZE,lHV)
```

The arguments in the call statement are defined as follows:

SZE = a floating point variable or constant equal in value to the desired size of the characters to be plotted, expressed in tenths of inches.

lHV = an integer variable or constant. If lHV = 0, characters are drawn parallel to the X-axis. Otherwise, characters are drawn parallel to Y-axis.

The FORMAT statement supplies the information as to what is to be drawn, and under what FORMAT specifications. The items in the FORMAT statement are defined as follows:

S = a FORMAT specification. FORMAT specifications may not be separated by a slash (/).

n = a statement number, corresponding to n in the WRITE statement.

The 100X is necessary for regulating record length on disk.

The WRITE statement places the characters which are to be plotted onto disk so that CHAR can reference them in order to translate the characters into plotter commands.

For information on how to use the Greek alphabet and other special characters, see appendix B.

## VI. The Care and Feeding of the Plotter Programs

### A. Operating Instructions

The plotter routines are part of the system library, and the job control statements that are needed in order to use PLOT or CHAR are a cataloged procedure. This procedure is specified on the EXEC card in the following manner:

```
//STEP1 EXEC PLOTECLG -- To compile and execute a
                        FORTRAN program which uses
                        PLOT and/or CHAR.
```

```
//STEP1 EXEC PLOTELG  -- To execute a previously
                        compiled FORTRAN program
                        which uses the PLOT and/or
                        CHAR routine.
```

NOTE: PLOTECLG has the same functions as FORTECLG; and PLOTELG has the same function as FORTELG. There is no equivalent to FORTECDK or FORTEC. To compile a program using PLOT and CHAR, FORTEC can be used.

### B. Output to the Plotter Tape

When the program begins outputting onto the plotter tape, the message "PLOTING BEGUN" is written on the printer. When plotting is completed (i.e., after a call to PLOT (100)) the message "PLOTING COMPLETED" is printed.

The programmer may find, when looking over his printed output, that he does not want his results plotted out on the 1627 (for example, if he realizes that his plotting would be incorrect). In this case, ask the operator in the IBM 360 room for a CANCEL PLOT slip which is filled out and returned to him. Note that if PLOTING BEGUN has not been printed then there is no plotter output and nothing could be canceled. However, if PLOTING BEGUN has been printed, there will be plotter output even if PLOTING COMPLETED has not been printed.

### C. Tape Procedures

The plotting output is put onto a 7-track tape which is later plotted out on the IBM 1627 using the IBM 1620. The Computation Center has a limited number of 7-track tapes which can be borrowed for this purpose (see Howard Stiles, Room M123). On the 360 Run Request Form check Plotter Tape under output and put the number of your 7-track tape in the Tapes Required section.

It is suggested that those, other than students, who use these routines frequently buy their own 7-track tape. Also, they should sign up for their own 1620 time to plot the tape. A copy of the 1620 program for writing this tape onto the plotter will be left in the right-most hopper on top of the 1622 card reader punch. Operating instructions for this program are under TAPE-TO-PLOTTER in the 1620 Systems Operating Instructions.

APPENDIX A

## CONCENTRIC CIRCLE ROUTINE

C  
C  
C

```

    DIMENSION ARRAY(11)
    DO 109 N=1,11
    READ(1,105) ARRAY(N)
105  FORMAT(12)
109  CONTINUE
    THETA=0.0
    IC=1
    ICC=90
    ICC=7
    T=61.
    XX=-5.3
    S=30.
    R=5.0
    M=T
    LL=1
2   CALL PLOT(IC,-5.0,5.0,5.0,1.0,-5.0,5.0,5.0,1.0)
    DO 106 N=1,11
    CALL PLOT(ICC,XX-5.4)
    WRITE(4,100)ARRAY(N)
    CALL CHAR(0.1,0)
100  FORMAT(12,100X)
    CALL PLOT(99)
106  XX=XX+1.
    CALL PLOT(ICC,5.2,-5.0)
    WRITE(4,101)
    CALL CHAR(0.1,0)
101  FORMAT(6HX-AXIS,100X)
    YY=5.0
3   CALL PLOT(ICC,-5.7,YY)
    IY=YY
    WRITE(4,103)IY
    CALL CHAR(0.1,0)
103  FORMAT(13,100X)
    YY=YY+1.0
    IF(YY-6.0)3,4,4
4   CALL PLOT(ICC,-7.2,0.0)
    WRITE(4,104)
    CALL CHAR(0.1,0)
104  FORMAT(6HY-AXIS,100X)
11  ICC = 10
    GO TO 24
12  ICC = 19
    GO TO 24
13  ICC = 20
    GO TO 24
14  ICC = 29
    GO TO 24
15  ICC = 30
    GO TO 24

```



```

16  ICC = 39
    GO TO 24
17  ICC = 40
    GO TO 24
18  ICC = 49
    GO TO 24
19  ICC = 50
    GO TO 24
20  ICC = 59
    GO TO 24
21  ICC = 0
    GO TO 24
22  ICC = 9
    GO TO 24
23  ICC = 90
24DO 25 K=1,M
    X=5.0
    X=R*COS(THETA)
    Y=R*SIN(THETA)
    THETA=THETA+3.14159265/S
    CALL PLOT(ICC,X,Y)
25  CONTINUE
    CALL PLOT(99)

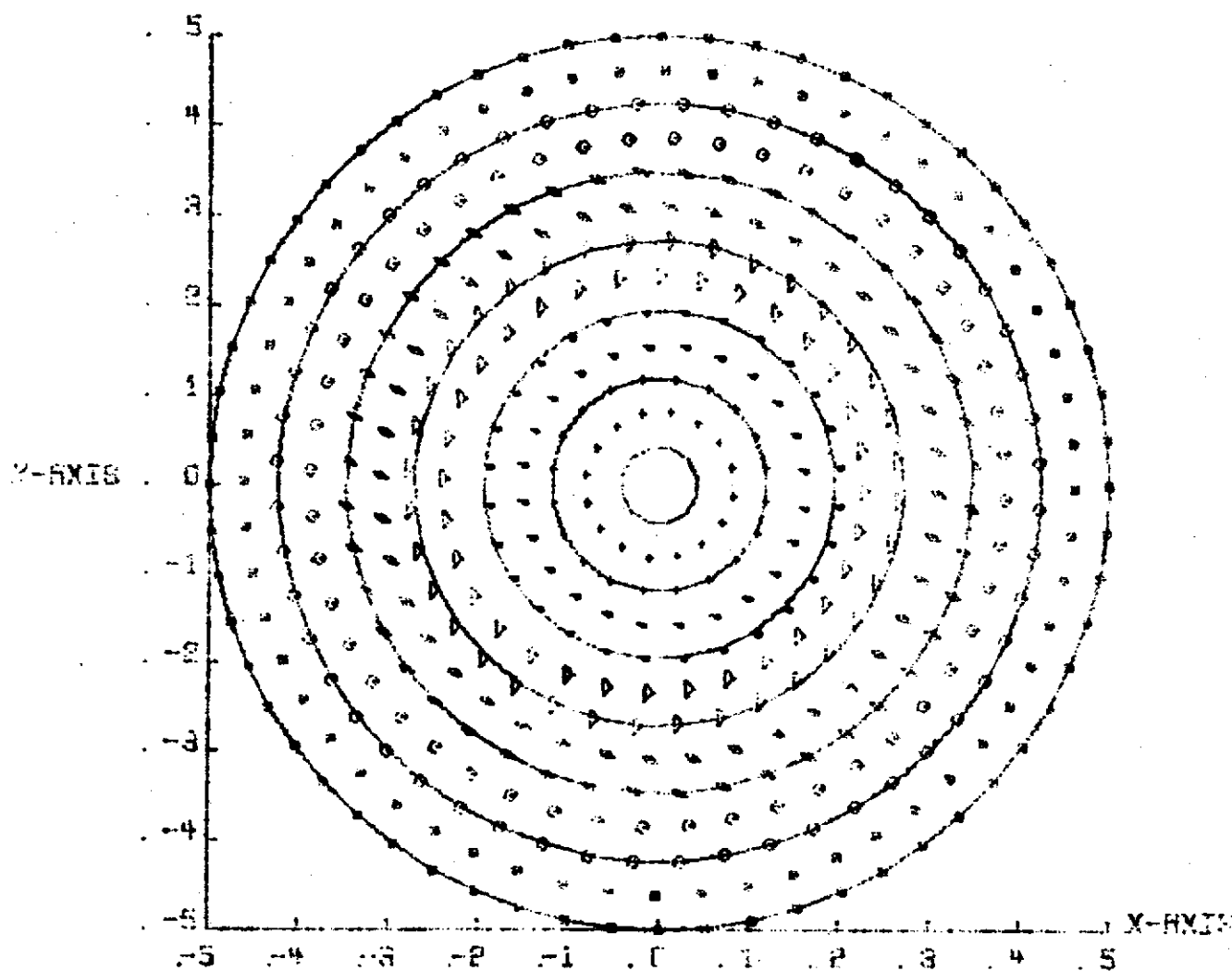
    M=T
    S=S+2.
    T=2.*S+1.
    R=R-.38
    LL=LL+1
    GO TO (11,12,13,14,15,16,17,18,19,20,21,22,23,99),LL
99  CALL PLOT(99)
    CALL PLOT(ICC,-10.,8.4)
    WRITE(4,200)
    CALL CHAR(0.4,
200  FORMAT(24H/STUVWXYZJKLMNOPQRSTUVWXYZ,100X)
    CALL PLOT(99)
    CALL PLOT(ICC,-10.,7.2)
    WRITE(4,201)
    CALL CHAR(0.4,0)
201  FORMAT(27HABCDEFGHJKLMNOPQRSTUVWXYZ,,100X)
    CALL PLOT(99)
    CALL PLOT(ICC,-10.,6.0)
    WRITE(4,202)
    CALL CHAR(0.4,0)
202  FORMAT(27H0123456789,R((,,'/.)*=Z.$*-,,100X)
    CALL PLOT(100)
    CALL EXIT
    END

```

5  
4  
3  
2  
1  
0  
-  
1  
2  
3  
4  
5

α β γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ σ τ υ φ χ ψ ω  
 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z  
 0 1 2 3 4 5 6 7 8 9 > < [ ] ≠ ± % / + (=) ° . \$

REPRODUCIBILITY OF THE  
 ORIGINAL PAGE IS POOR



APPENDIX B

## Plotting Special Characters

It is possible in CHAR to use the Greek Alphabet and the special symbols: <, >, ≠, %, ¢, ], [, °. This is accomplished by multipunching the FORMAT statement characters in accordance with the following symbol table. Some confusion may arise when the programmer looks at how this FORMAT statement is compiled as these special multipunched characters may cause totally unrelated characters to print out during compilation.

Example 1: If one wishes to plot  $\alpha = \beta + \gamma$   
the FORMAT statement would be punched

```

      12      12      12
FORMAT(5H  0 =  0 +  0) or
          1      2      3

```

```

      12      12      12
FORMAT ( 0 =  0 +  0)
          1      2      3

```

NOTE: The ' mark in the second example is a 5/8 punch

Example 2: Plot  $\delta = 50^\circ$ ,  $A < B$ ,  $\phi = [(RADIUS) * \pi]$

```

      12      11      12      11      11      12 11 5
      5 12      11      12      11      11      11 0 8)
FORMAT (8  0 = 50  0, A  8, B  0 =  0 (RADIUS)* 7  9
          4      9      4      4      8
          8
          2

```

In the following table a ✓ next to a character indicates that this is the standard keypunch code for the 026 keypunches so that these characters need not be multipunched. In the case where there is more than one multipunch defined for a character, either multipunch can be used.

## APPENDIX B-2

CHARACTER	IBM		PUNCH		CHARACTER	IBM		PUNCH
	0 2 6	0 2 9				0 2 6	0 2 9	
α			12-0-1		A	✓	✓	12-1
β			12-0-2		B	✓	✓	12-2
γ			12-0-3		C	✓	✓	12-3
δ			12-0-4		D	✓	✓	12-4
ε			12-0-5		E	✓	✓	12-5
ζ			12-0-6		F	✓	✓	12-6
η			12-0-7		G	✓	✓	12-7
θ			12-0-8		H	✓	✓	12-8
ι			12-0-9		I	✓	✓	12-9
κ			12-11-1		J	✓	✓	11-1
λ			12-11-2		K	✓	✓	11-2
μ			12-11-3		L	✓	✓	11-3
ν			12-11-4		M	✓	✓	11-4
ξ			12-11-5		N	✓	✓	11-5
ο			12-11-6		O	✓	✓	11-6
π			12-11-7		P	✓	✓	11-7
ρ			12-11-8		Q	✓	✓	11-8
σ			12-11-9		R	✓	✓	11-9
τ			11-0-2		S	✓	✓	0-2
υ			11-0-3		T	✓	✓	0-3
φ			11-0-4		U	✓	✓	0-4
χ			11-0-5		V	✓	✓	0-5
ψ			11-0-6		W	✓	✓	0-6
ω			11-0-7		X	✓	✓	0-7
					Y	✓	✓	0-8
					Z	✓	✓	0-9
ν		✓	0-8-6				✓	
^			11-8-2					
⌈			12-11-0-9-8-4		0	✓	✓	0
⌋			12-11-0-9-8-5		1	✓	✓	1
#			12-11-0-9-8-3		2	✓	✓	2
%			12-11-0-9-8-6		3	✓	✓	3
&			8-7		4	✓	✓	4
/	✓	✓	0-1		5	✓	✓	5
+	✓		12		6	✓	✓	6
+	✓	✓	12-8-6		7	✓	✓	7
(	✓		0-8-4		8	✓	✓	8
(	✓	✓	12-8-5		9	✓	✓	9
=	✓		8-3					
=	✓	✓	8-6		o			
)	✓		12-8-4		(Degree)			12-11-0-9-8-2
)		✓	11-8-5		.	✓	✓	12-8-3
					\$	✓	✓	11-8-3
					*	✓	✓	11-8-4
					-	✓	✓	11
					,	✓	✓	8-3-0

APPENDIX C

## Common Error Messages

<u>Messages</u>	<u>Description</u>
1HC2121	Check that the FORMAT statements for CHAR have 100X in them.
INVALID IC = --	In the call to PLOT, IC was set equal to an invalid code (see pages 2-3) or was not a fixed-point number.
Blank plotted on output where character was expected.	The programmer attempted to plot (using CHAR) an illegal character. See Appendix B.
1HC2101	Illegal arguments passed in PLOT or CHAR. (For example, if X or Y had been fixed point variables.)

## ADDITIONAL IC-CODES (91 and 88) FOR THE PLOT SUBROUTINE

1. An IC-code of 91 is practically identical to IC = 90, except that with a 91 code, the pen remains in the 'UP' position.
2. A size parameter specification has been added to the PLOT subroutine. To make use of this option, the programmer must employ the following call statement:

CALL PLOT (88,X,Y,M,N)

This option should not be used for the following codes:

100, 99, 91, 90, 7, 1, 101, and 201

The above numbers should be used as first parameters in CALL PLOT.

An IC-code of 88 enables the programmer to state the size of the symbols to be drawn. X and Y are floating point variables for the coordinates X and Y; M indicates the symbol code as shown in the old PLOT instructions (see blue stencil) e.g., -- 29 for the octagon, leaving the pen up after the symbol is drawn on the plotter, -- 40 for a triangle leaving the pen down, etc.

N may be any whole number ranging from 1 through 79 to give the sizes of the desired symbol. The maximum size will be little over 1 1/2 inches (79 x .02). For instance, the statement CALL PLOT (88,X,Y,19,50) will cause a square of 1 inch high to be drawn in XY. XY is in the center of every symbol except for the octagon, in which case XY is off center:

A value greater than 79 for N will give an error message.

Below is a list of variables for the corresponding sizes (given at intervals of 5).

<u>N</u>	<u>SIZE</u> (of symbol in inches)
0	.0
5	.1
10	.2
15	.3
20	.4
25	.5 (1/2 in.)
30	.6
35	.7
50	1.0 (1 in.)
55	1.1
60	1.2
65	1.3
70	1.4
75	1.5 (1 1/2 in.)
min. 1	.02
max. 79	1.38

APPENDIX D

## Storage Requirements

The PLOT and CHAR subroutine package is composed of the following routines:

<u>Name</u>	<u>Length</u>	<u>Use In</u>
PLOT	AB0	PLOT
CHAR	2C8	CHAR
CHAR	884	CHAR
ØPT	614	PLØT and CHAR
BAD	1D0	PLØT
LAST	218	PLØT
CTØ	32	PLØT and CHAR

$$\text{Total Length} = 205A_{16} = 8282_{10}$$

Minor changes may be added to these programs, thus altering their length slightly.



## Section 8

### Response Function Programs

	<u>Page</u>
I. Introduction	8-1
II. Macro Systems Flowchart	8-2
III. OSOH100	8-3
IV. OSOH101	8-4
V. OSOH102	8-5
VI. OSOH103	8-6
VII. OSOH105	8-8
VIII. OSOH105A	8-9
IX. OSOH106	8-10
X. OSOH108	8-11
XI. OSOH109	8-13
XII. OSOH110	8-14
XIII. OSOH118	8-15
XIV. OSOH120	8-16
XV. Subroutine FGET (IDSN,IRSLT,IOVEC)	8-17
XVI. Subroutine FPUT (IDSN,ICMD,IRSLT,LQVEC)	8-19
XVII. Description of Final Output Tape	8-21

## I. Introduction

Five 7-track magnetic tapes generated at BBRC and containing the calibration data were merged and converted into a 9-track 360-compatible tape. The data set on this tape was compared with the log book records and the erroneous records were corrected.

Since the gain of the system changed by about 7% during the calibration period, gamma ray lines 1.460 MeV ( $K^{40}$ ) and 2.615 MeV ( $Th^{228}$ ) observed in the background spectra were used to determine gain as a function of time over the 5-day period. Background data averaged over ten-minute intervals distributed throughout the calibration period was used to determine  $n_0$  and C. The value for  $n_0$ , averaged over the entire calibration period, was then used to determine C for individual graphs; linear interpolation was used to find C for intermediate times.

After determining C for each data set, the spectra were then smoothed and the gain corrected to a standard C ( $= 0.444 \times 10^{-4}$ ) in the manner described by P. Higbie, D. Forrest, I. Gleske, E. Chupp, Experimental Gamma Ray Response Function for the OSO-7 Spacecraft, Nucl. Instr. Meth. 108 (1973), 167-176.

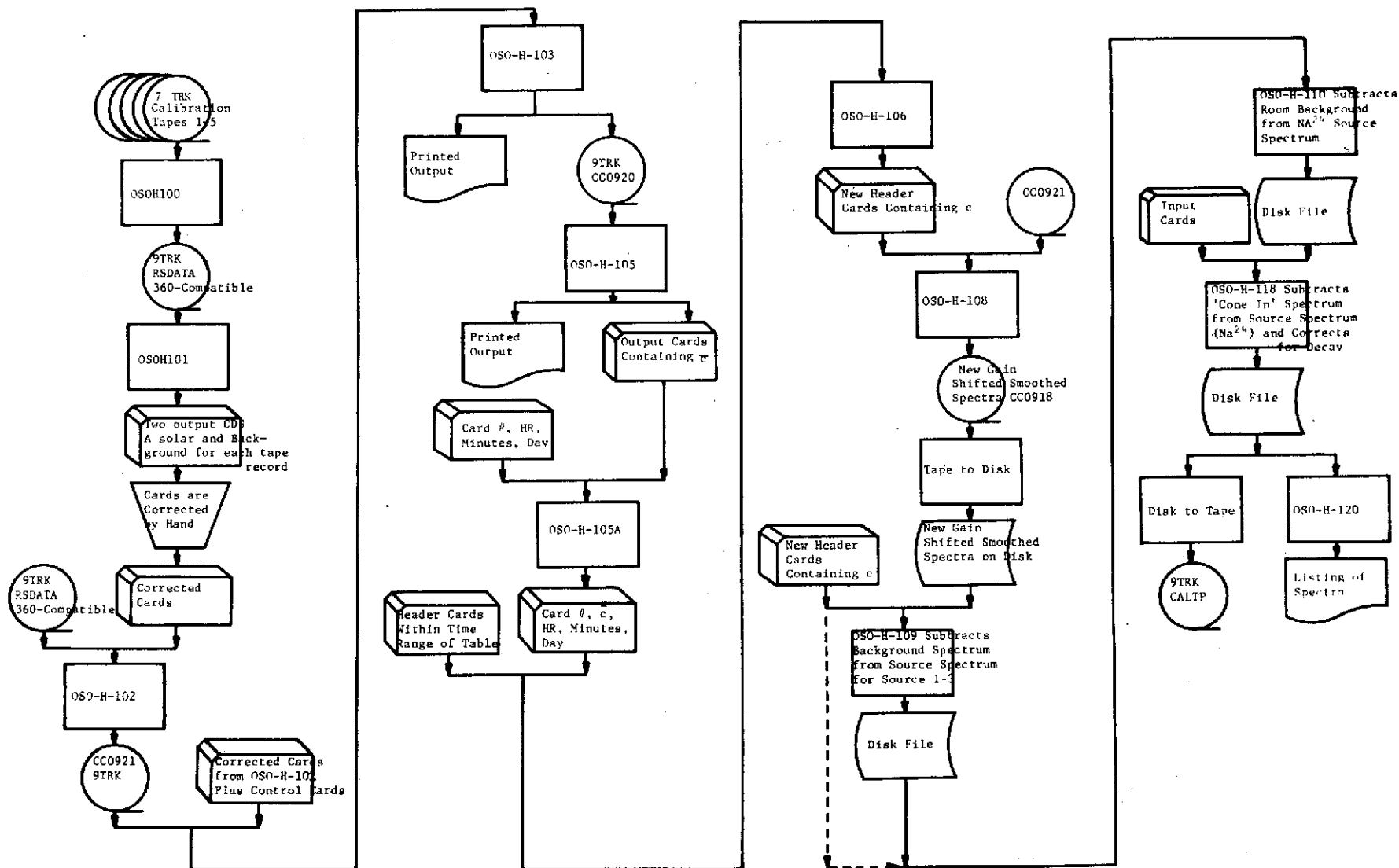
Data sets with the same elevation ( $\theta$ ) and azimuth ( $\phi$ ) angles were added together and average spectra for each set of  $\theta$  and  $\phi$  determined;  $Na^{24}$  spectra were treated differently from the other three sources. In the case of  $Na^{24}$  spectra, room background was first subtracted from the source spectra. This was done to remove background contribution due to natural radioactivity (mainly  $K^{40}$  and  $Th^{228}$ ).

The room background-corrected spectra were then corrected for the activity difference between the two  $Na^{24}$  sources and decay in the counting rate due to short half-life (15 hours) of the  $Na^{29}$  source. [Two series of runs were made separately with the two sources listed in Higbie et al. (1973).]

The runs with the shadow cone 'in' were then subtracted from the corresponding spectra with the shadow cone 'out'. This provided a rough correction for the effect of scattering and room background correction for  $Sn^{113}$ ,  $Cs^{137}$ , and  $Zn^{65}$  sources.

Final spectra for the different values of  $\theta$  and  $\phi$  are on tape CALTP.

# 11. MACRO SYSTEMS FLOWCHART



## III. OSOH100

Program Language : Fortran IV G-Level  
Programmer : S. Croteau  
Installation : University of New Hampshire  
Computer : IBM-360-H50  
Subprograms Called : FGET, SHIFT, FPUT

This program reads the calibration data tapes generated at BBRC using FGET with data converter off, calls SHIFT to reorganize the data bits so that two bytes of data are derived from every three bytes read, and writes the new records onto a 9-track tape using FPUT.

## IV. OSOH101

Program Language : Fortran IV G-Level  
Programmer : J. Kantauskis  
Installation : University of New Hampshire  
Computer : IBM-360-H50  
Subprograms Called : FGET

This program reads the 9-track 360-compatible calibration tape called RSDATA. From each tape record a background card and a solar card is punched. OSOH101 right justifies THETA and PHI and converts THETA, PHI, and TIME from the Varian Data 620/i internal octal code to character code. OSOH101 also converts the source field to a number code. This is done by checking column 21 of the tape record for an S( $\text{Sn}^{133}$ ), C( $\text{Cs}^{137}$ ), Z( $\text{Zn}^{65}$ ), N( $\text{Na}^{24}$ ), B (background in Varian octal code) and converting it to a 1, 2, 3, 4, or 5, respectively. The program also searches the cone field of the tape and converts the C (cone) or N (no cone) from octal format to character format.

## V. OSOH102

Program Language : Fortran IV, G-Level  
Programmer : J. Kantauskis  
Installation : University of New Hampshire  
Computer : IBM-360-H50  
Subprograms Called : FGET, FPUT

OSOH102 reads the 9-track 360-compatible calibration data tape (RSDATA) and the corrected header cards from OSOH101. For each tape record read two header cards (background and solar) are read. Each header card is tested for a C in the cone field. If there is a C in the cone field, a new cone field is given a value of '1' otherwise the new cone field is given a value of '2'. Characters 42 and 426 of the tape are the status fields. These two fields are converted from octal to binary. The background header (the first of the two headers read) and the 11th and 12th binary positions of the status plus characters 41-424 of RSDATA are used to create a background record on the new tape. The same is done to create each solar record except the solar header and characters 425-808 of RSDATA are used.

## VI. OSOH103

Program Language : Fortran IV, G-Level  
Programmer : J. Kantauskis  
Installation : University of New Hampshire  
Computer : IBM-360-H50  
Subprograms Called : MATCH1, MATCH2

Before OSOH103 is executed, control cards are inserted manually into the header deck (from OSOH101) at the desired locations. The three control cards and their meanings are:

- 1) '7' in column 1 : Called ACODE in program. ACODE means add the corresponding data fields of the tape which match the following headers, separating background records from solar records.
- 2) '8' in column 1 : Called SCODE in program. SCODE means add the corresponding data fields of the tape records which match the following headers accumulating background and solar data separately.
- 3) '9' in column 1 : Called ECODE in program. When ECODE is read, it means execute. There are two cases:
  - a) ECODE follows directly behind the ACODE group of headers. Here ECODE means to graph the sum of the solar and sum of the background records which were accumulated beforehand.

- b) ECODE follows directly behind the SCODE group of headers. Here ECODE means to graph the sum of the background and solar records which were accumulated after SCODE. Then to subtract the sum of the background and solar sums taken after SCODE. Then to graph this difference.

The header cards are read and matched with the records on the tape generated by OSOH102.



## VII. OSOH105

Program Language : Fortran IV, G-Level  
Programmer : J. Kantauskis  
Installation : University of New Hampshire  
Computer : IBM-360-H50  
Subprograms Called : None

This program reads the tape generated by OSOH103 and calculates  $n_0$  and  $\bar{C}$ . The program then punches the 122 values of  $\bar{C}$  along with the card number.

## VIII. OSOH105A

Program Language : Fortran IV, G-Level  
Programmer : J. Kantauskis  
Installation : University of New Hampshire  
Computer : IBM-360-H50  
Subprograms Called : None

This program combines two decks of cards to form one deck. It reads one deck containing card number, hours, minutes, and day, and another deck containing card number and  $\bar{c}$  (output of OSOH105). It then punches cards containing card number,  $\bar{c}$ , hours, minutes, and day.

## IX. OSOH106

Program Language : Fortran IV, G-Level  
Programmer : J. Kantauskis  
Installation : University of New Hampshire  
Computer : IBM-360-H50  
Subprograms Called : None

This program reads the 122-card table created by OSOH105A. Once the table has been read in, header cards, which fall within the time range of the table, are read. For each header card the appropriate tabular entries are determined based on 'day' and 'time'. Then 'c' is determined for the header card by interpolating between 'c' and 'time' of the table and 'time' of the header card.

$$C = C_1 + \frac{T - T_1}{T_2 - T_1} (C_2 - C_1)$$

Finally the new header cards are punched which will now contain the new values of 'c'.

The program OSOH106A is a version of OSOH106 with a change in the format statement of the input card and output card as shown in the comment cards in program OSOH106.

## X. OSOH108

Program Language : Fortran IV, G-Level  
 Programmer : J. Kantauskis  
 Installation : University of New Hampshire  
 Computer : IBM-360-H50  
 Subprograms Called : FGET, SMOTHE, GAIN, FPUT

This program reads the header cards finds the matching records on tape (output of OSOH102), smooths the spectra, then gain shifts the smoothed spectra and writes the smoothed shifted spectra on another tape. The program also keeps a total of the counts in the smoothed shifted spectra.

## Smoothing Subroutine:

The function used to smooth the spectra is the following:

$$Y_i = \frac{(-3Y_{i-2} + 12Y_{i-1} + 17Y_i + 12Y_{i+1} - 3Y_{i+2})}{35}$$

The proof and explanation of this can be found in Volume 36 Number 8 of the July 1964 issue of Analytical Chemistry, "Smoothing and Differentiation of Data by Simplified Least Squares Procedures", by Savitsky and Galay.

## Gain Shifting Subroutine:

$Y_1$  and  $Y_2$  are calculated as follows:

$$Y_1 = \sqrt{C(N_1 + N_0)^2 / C_s} - M_0$$

$$Y_2 = \sqrt{C(N_2 + N_0)^2 / C_s} - M_0$$

Where C is read from the header cards, the rest is given in the program.

$M_1$  is the next integer higher than  $Y_1$   
 $M_2$  is the next integer lower than  $Y_2$

Now the shifted channels  $X_i$  are found as follows:

$$X_i = \sqrt{C_S (i + M_0)^2 / C - N_0} \quad \text{where } i \text{ is the loop from } M_1 \text{ to } M_2.$$

The  $Y_i$  values corresponding to the  $X_i$  values are found through interpolation. The three interpolation methods are used, Newton's Forward, Newton's Backward, and Stirlings Central Difference. Once the values of  $Y_i$  have been found, the shifted spectrum  $M(i)$  is calculated from  $i$  to  $(M_2 - M_1 + 1)$

$$M_i = \frac{C_S}{C} \times \frac{(M + M_0)}{(X_i + N_0)} \times Y(X_i)$$

where  $M$  goes from  $M_1$  to  $M_2$ .

## XI. OSOH109

Program Language : Fortran IV, G-Level  
Programmers : J. Kantauskis, P. Higbie  
Installation : University of New Hampshire  
Computer : IBM-360-H50  
Subprograms Called : None

The tape that was generated by OSOH108 is read onto disk and remains on disk until the conclusion of this series of production runs. The major function of OSOH109 is to subtract the background spectrum (cone in) from the source spectrum for sources Sn, Zn, and Cs. The new records are written onto a new disk file.

## XII. OSOH110

Program Language : Fortran IV, G-Level  
Programmers : J. Kantauskis, P. Higbie  
Installation : University of New Hampshire  
Computer : IBM-360-H50  
Subprograms Called : None

This program reads both data sets on disk and continues the work of OSOH109 by writing more records on the new data set. Its major function is to subtract room background from the source spectrum for  $\text{Na}^{24}$ .

## XIII. OSOHL18

Program Language : Fortran IV, G-Level  
Programmers : J. Kantauskis, P. Higbie  
Installation : University of New Hampshire  
Computer : IBM-360-H50  
Subprograms Called : None

This program reads the new data set on disk, subtracts the 'cone in' spectrum from the source spectrum for  $\text{Na}^{24}$ , corrects for decay, and then writes the new record onto the same data set on disk.



## XIV. OSOH120

Program Language : Fortran IV, G-Level  
Programmers : J. Kantauskis, P. Higbie  
Installation : University of New Hampshire  
Computer : IBM-360-H50  
Subprograms Called : None

This program reads the file on disk and prints out the desired spectra.

## XV. Subroutine FGET (IDSN,IRSLT,IOVEC)

Program Language : IBM O/S 360 Assembler Language

Date Written : February 4, 1971

Programmer : A. Buck

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments : 1) IDSN: a full word input integer containing the data set number to which the call is directed. (IDSN must be 1, 2, or 3).

2) IRSLT: a full word output integer in which subroutine FGET may place an activity-result code as follows...

a) IRSLT = -1 if the requested activity was invalid. This code is returned if either of the following occur:

i. IDSN contains a number less than 1 or greater than 3.

ii. Data set has been previously closed.

b) IRSLT = 0 if data is read and no I/O error is detected.

c) IRSLT = 1 if an I/O error occurred on the block containing the data placed in IOVEC. The calling program has the responsibility for accepting or rejecting the data. In addition, the calling program is responsible for proper reaction to the remaining records in the block for which IRSLT will be 0.

d) IRSLT = 2 if end-of-file was detected. The contents of IOVEC are not changed. The data set is closed and can not be used again by subroutine FGET with the same value of IDSN.

3) IOVEC: an output vector in which FGET can place the record. The actual variable used for IOVEC may be integer, real, logical, or complex. The first leftmost byte of the data record is placed in the high-order byte of IOVEC when reading occurs. The number of bytes placed in IOVEC is specified by the LRECL subparameter of the DD Statement.

Subprograms Called : None

Description : This subroutine is a QSAM input subroutine to read logical records into an area defined by the calling program. This routine is specifically designed to supply a Fortran calling program with logical records from a data set to which the DCB subparameter 'RECFM = FB' applies. For data sets to which 'RECFM = F' or 'RECFM = U' applies, subroutine FTREAD may be more easily used. 'RECFM = V' is specifically prohibited.

Subroutine FGET can manage as many as 3 data sets during the same job step. The data sets may be open (active) simultaneously provided no two data sets are on the same magnetic tape.

DD Statements required...

If FGET is to be called with IDSN = 1 (or 2 or 3), then the user must describe the data set by a DD Statement having the DD name FGET01 (or FGET02 or FGET03). The DD Statement must include the DCB subparameters 'RECFM', 'LRECL', and 'BLKSIZE' but must omit 'DSORG' or 'MACRF'.

## XVI. Subroutine FPUT (IDSN, ICMD, IRSLT, IOVEC)

Program Language : IBM O/S 360 Assembler Language

Date Written : February 4, 1971

Programmer : A. Buck

Installation : University of New Hampshire  
Space Science Center

Computer : IBM-360-H50

Dummy Arguments : 1) IDSN: a full word integer input parameter specifying the data set to which activity is directed. The integer stored at IDSN must be greater than zero but less than 21.

2) ICMD: a full word input integer variable containing a request type code which specifies the activity to be performed.

a) ICMD = 1 implies that a logical record is to be written.

b) ICMD = 2 implies the data set is to be ended. Once ended, the data set is unavailable to the calling program through the same data set number.

3) IRSLT: a full word output integer variable in which subroutine FPUT can place a return code as follows

a) IRSLT = -1 implies request was illegal for one of the reasons below:

i. IDSN contained a value less than 1 or greater than 20.

ii. ICMD contained a value less than 1 or greater than 2.

iii. A write request was made to a data set that had been ended.

iv. An endfile request was made to a data set either before the first write request or after the data set had been ended.

b) IRSLT = 0 implies request was executed normally.

c) IRSLT = 1 implies an I/O error occurred while writing a block. The data set was ended.

4) IOVEC: an input vector whose high order byte is the first byte to be written (when ICMD = 1) in a logical record.

The number of bytes written is specified by the LRECL subparameter on the DD statement describing the data set.

Subprograms Called : None

Description : Subroutine FPUT is a QSAM output subroutine to write logical records from an area defined by the calling program. This routine is specifically designed to allow a Fortran program to create without format control an extensive data set to which the DCB subparameter 'RECFM = FB' applies. For data sets to which 'RECFM = F' or 'RECFM = U' applies, subroutine FTWRIT may be more easily used.

Subroutine FPUT can manage as many as 20 data sets in the same job step. As many as desired may be active simultaneously. Once the endfile command has been issued to a data set, subroutine FPUT will refuse to perform any further activity for that particular data set. The large number of data sets allows writing as many as 20 files to the same magnetic tape volume as a possible application.

DD Statements required...

If subroutine FPUT is called with IDSN=1 (or 2, ..., or 20) the user must supply a DD Statement having the DD name FPUT01 (or FPUT02, ..., or FPUT20). The DD Statement must include the DCB subparameters 'RECFM', 'LRECL', and 'BLKSIZE' but must omit 'DSORG' and 'MACRF'.

## XVII. Description of Final Output Tape

The data control block for this tape is as follows:

DCB = (RMCFM = FB, LRECL = 1680, BLKSIZE = 1680)

The tape contains the final spectra for the different values of  $\theta$  and  $\phi$ . All variables on this tape are 32-bit words. the format is as follows:

Word  
No.

1	: Source number	(integer)
2	: Elevation angle ( $\theta$ )	(integer)
3	: Azimuth angle ( $\phi$ )	(integer)
4	: First channel ( $M_1$ )	(integer)
5	: Last channel ( $M_2$ )	(integer)
6	: Sum of counts in the original source spectrum	(floating point)
7	: Sum of counts in the original background spectrum	(floating point)
8	: Sum of counts in the smoothed gain-shifted source spectrum from $M_1$ to $M_2$	(floating point)
9	: Sum of counts in the smoothed gain-shifted background spectrum from $M_1$ to $M_2$	(floating point)
10	: Same as 8 but accumulated in a different step	(floating point)
11	: Same as 9 but accumulated in a different step	(floating point)
12-20	: Counters and extra words	(floating point)
21-420	: Spectrum (400 channels)	(floating point)

Note: In the  $\text{Na}^{24}$  source, Nos. 6-11 appear in the following order:  
6, 8, 10, 7, 9, 11.

OSO-H  
EXPERIMENT, DESCRIPTION AND OPERATION  
UNH GAMMA RAY MONITOR

Principal Investigator:

E. L. Chupp

Co-Investigators:

P. R. Higbie  
A. A. Sarkady

University of New Hampshire  
Department of Physics  
Durham, New Hampshire 03824

(Revision Date: June 2, 1970)

This report is prepared in response to Goddard Space Flight  
Center request of 15 July 1969; File 838.

## TABLE OF CONTENTS

	<u>Page</u>
I. EXPERIMENT DESCRIPTION.....	1.
A. Objective.....	1.
B. Instrument.....	2.
II. EXPERIMENT OPERATION.....	11.
III. DATA FORMAT.....	17.
A. Main Frame Data.....	17.
B. DSF Data.....	24.
C. Analog Subcom Data.....	24.
IV. SPECIAL HANDLING REQUIREMENTS.....	35.
V. ENVIRONMENTAL REQUIREMENTS.....	35.
VI. SPECIAL TEST REQUIREMENTS AT BBRC & ETR...	36.
VII. ALIGNMENT REQUIREMENTS.....	36.
VIII. RADIOACTIVE SOURCE REQUIREMENTS.....	36.
IX. HIGH VOLTAGE OPERATION REQUIREMENTS.....	37.
X. SQUIB REQUIREMENTS.....	37.
XI. TEST PROCEDURES AT BBRC.....	38.

LIST OF FIGURES

LIST OF TABLES



## I. EXPERIMENT DESCRIPTION

### A. Objective:

The primary objective of the proposed experiment is to monitor the high energy photon spectrum (0.3 to 9.1 MeV) at the OSO altitude (approximately 575 km) and correlate the observations with the time when the detector is exposed to the sun and with other observations of solar activity. Our primary interest is to study the characteristics (intensity and time variation) of gamma rays emitted by the sun. If nuclear reactions occur in the solar atmosphere, either continuously or sporadically, then several nuclear gamma rays might be produced in sufficient quantity to be observable at the earth. An important feature of this gamma ray experiment is the measurement of gamma ray energies with the highest resolution possible with the basic detector and the spacecraft telemetry system in order to emphasize any line structure in the gamma ray flux. As an example, gamma rays at 0.51 MeV, 2.22 MeV, 4.43 MeV, and 6.15 MeV would indicate production of positrons, neutrons, and excited nuclei from various nuclear processes.

The experiment will also provide monitoring of the continuous spectrum of photons in the energy range from 0.3 to 9.1 MeV. Additional information from the experiment will provide knowledge of changes in background radiations. These are likely due to earth albedo gamma rays, electron bremsstrahlung, and neutron and charged particle interactions in the

detector and the spacecraft. This information is important for further understanding of solar-terrestrial relations since some gamma rays may be produced in the earth's atmosphere by other solar radiation. Also, such information is essential for determining the direction for detector development for solar and celestial gamma ray astronomy.

The main features of the experiment and detector may be summarized as follows:

(1) Monitor solar gamma ray flux from 0.3 to 9.1 MeV with a high resolution gamma ray spectrometer to search for nuclear gamma ray lines.

(2) Monitor the continuum solar gamma ray spectrum over the same energy range, and simultaneously monitor the 7.5 to 120 keV X-ray range.

(3) Provide for internal calibration of detector automatically or on external command.

(4) Data handling mode may be changed on command to emphasize high resolution or a broad spectral survey.

#### B. Instrument:

The gamma ray spectrum is measured by means of a high resolution NaI(Tl) gamma ray spectrometer surrounded with a CsI(Na) active collimator which also acts as a charged particle shield. The basic layout of the detector is shown in Figure 1. The central gamma ray spectrometer is viewed by a single ruggedized photomultiplier. The CsI(Na) shield, which consists of a large cup and a front slab, is viewed by five photomultipliers (RCA70132B-Hemispherical Photocathode), three

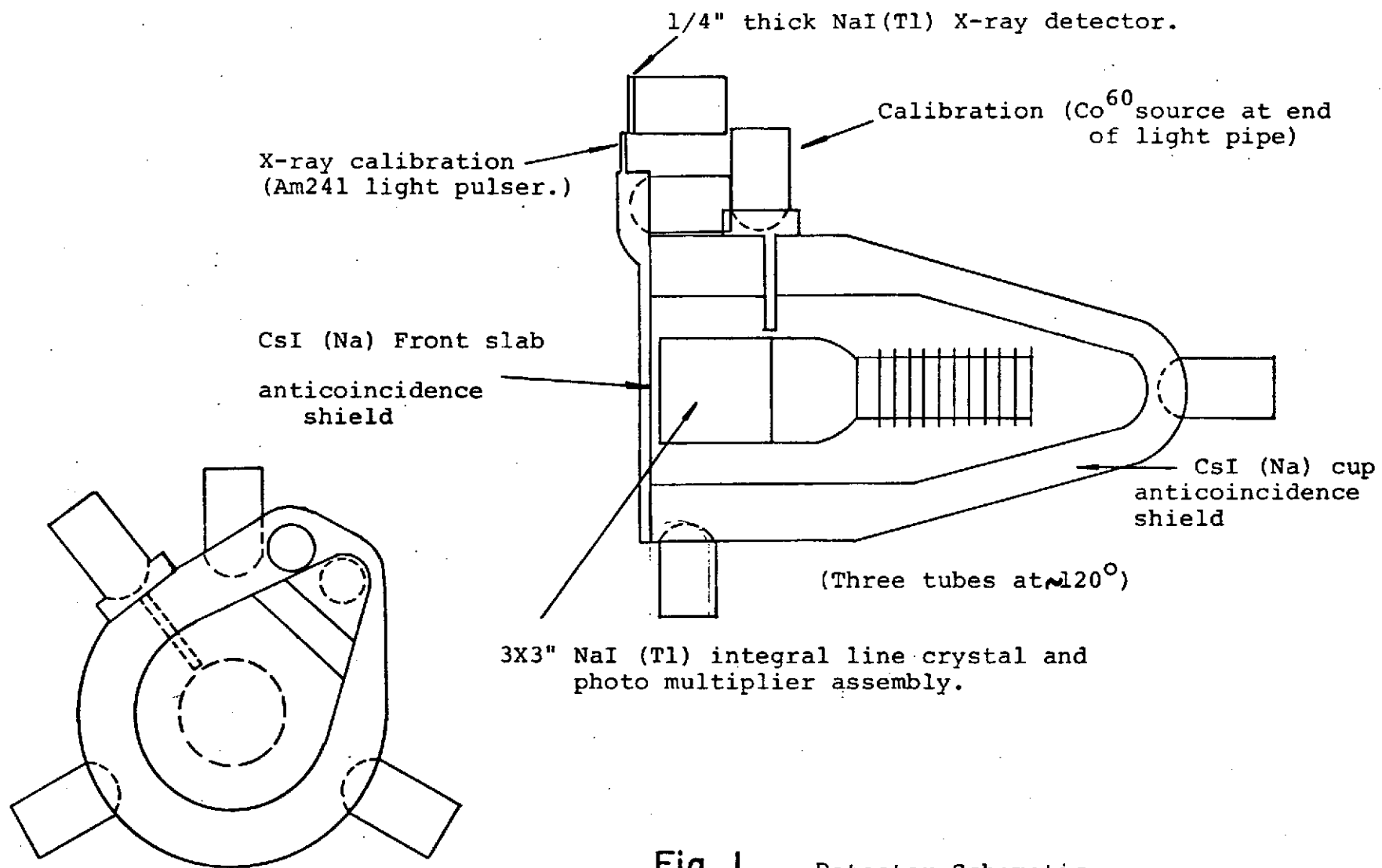


Fig. 1 Detector Schematic

of which are arranged around the front end and one at the rear end of the cup, and the fifth views the front slab of the shield and an  $\text{Am}^{241}$  calibration source. In addition, a single photomultiplier (RCA 4461) views a thin  $\text{NaI(Tl)}$  crystal (1/4"), which acts as a detector for low energy solar X-rays and low energy charged particles. The purpose of the detector is to provide data to be used for correlation purposes in the data analysis. Another PM tube (RCA70132B) views a calibration source  $\text{Co}^{60}$  (see below).

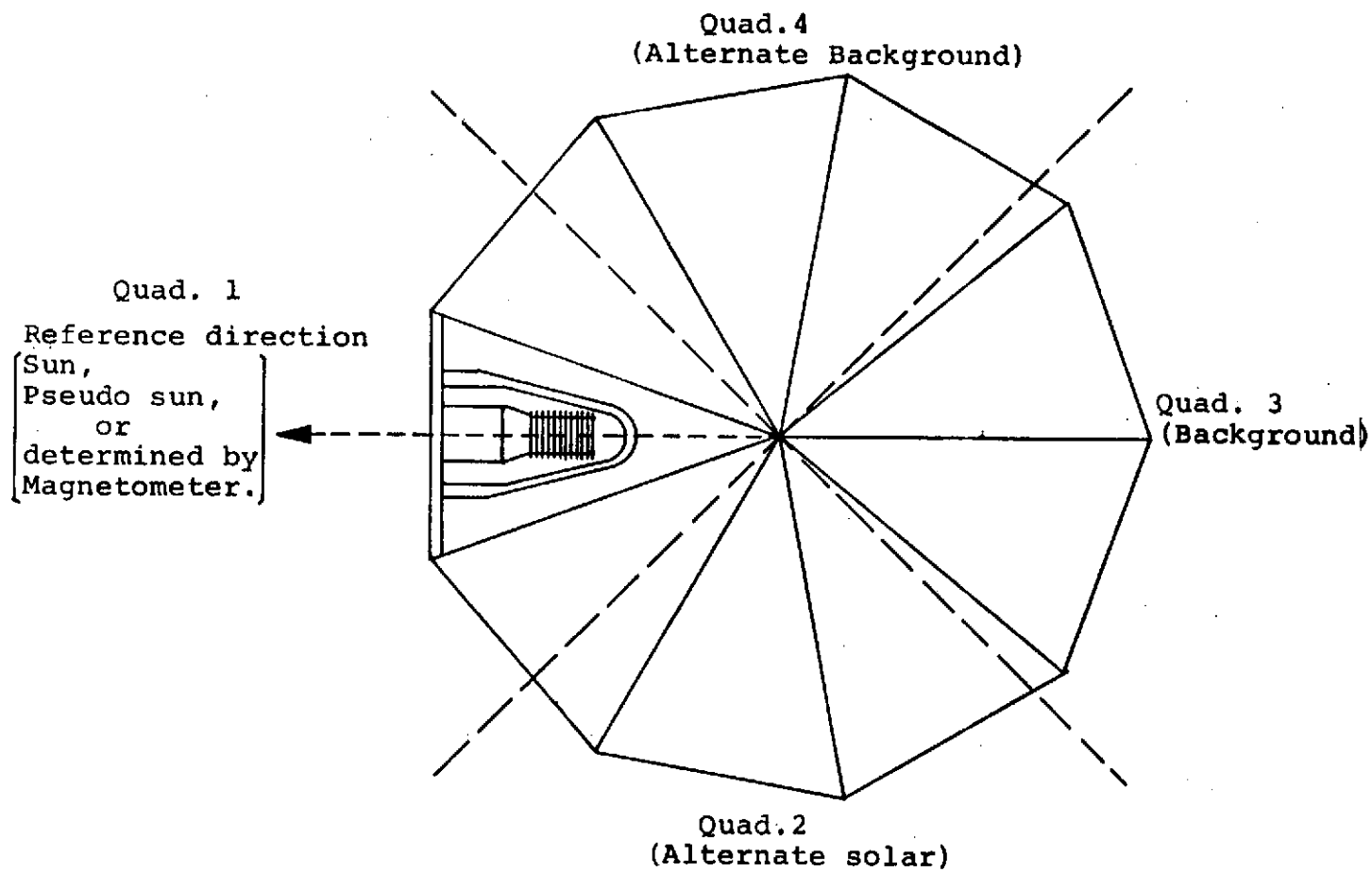
The arrangement of the detector in the OSO wheel compartment (#9) is shown in Figure 2. Quadrants 1-4 give the various viewing aspects for the detector relative to the spacecraft sun line. These will be referred to again in the discussion of the experiment operation.

Figure 3 shows the system functional block diagram. Basic scientific data is represented by:

- (1) the gamma ray spectrum from the central detector as measured by the 377 channel quadratic pulse height analyzer,
- (2) the X-ray spectrum over the energy range 7.5-120 keV as measured by a four channel pulse height analyzer,
- (3) high energy neutron events occurring in the central detector as measured by pulse shape discrimination circuitry.

Additional detector housekeeping data monitored is:

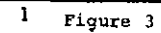
- (1)  $\text{CsI(Na)}$  front slab counting rates
- (2)  $\text{CsI(Na)}$  cup counting rates
- (3) Central detector integral counting rates



View from top at instant compartment C/L lies along sun line.

Fig. 2

2



Calibration of the spectrometer and X-ray detector is accomplished periodically by use of a  $\text{Co}^{60}$  source giving gamma lines at 1.17 and 1.33 MeV, and an  $\text{Am}^{241}$  source giving an X-ray line at 60 keV and iodine escape peak at 32 keV and possible  $\text{Np}^{237}$  X-rays. Each of these sources consists of a scintillating material viewed by one of the photomultipliers. Since the gamma rays or X-rays from the sources are coincident with the charged particles emitted by the sources, the calibration lines may be electronically rejected or selected. Electronic calibration signals are also provided. The calibration data appears on the normal data channels but is identified by an analog word ASSC 1-4 and bits 5 & 6 in the MF status word.

The assignment of scientific and housekeeping data is given in Section III and also Table I, Data Format. The assignment of commands is given in Table II. The command number assignment agrees with that given in the UNH interface pin list (Table 8-1, page 8-2 of the Preliminary Allocation and Performance Summary Data dated June 15, 1969).

TABLE I (A)  
DATA ASSIGNMENT

Measurement	Time resolution	Energy and Item	Readout Assignment
Central Detector	184 Sec. 61 Sec. 30 Sec.	377 ch, .3 to 9.1 MeV solar-background 121 ch, .3 to 9.1 MeV solar-background 57 ch, .3 to 9.1 MeV solar-background	2 MF word (11,12)
	15 Sec.	.3 to 9.1 MeV B .3 to 9.1 MeV S	ASSC 3 ASSC 7
	15 Sec.	>9.1 MeV B >9.1 MeV S	ASSC 4 ASSC 8
X-Ray Detector	30 Sec.	7.5-15 keV solar 7.5-15 keV background 15-30 keV solar 15-30 keV background 30-60 keV solar 30-60 keV background 60-120 keV solar 60-120 keV background	DSM 23 $t_n$ DSM 29 $t_n$ DSM 47 $t_n$ DSM 9 $t_n$ DSM 23 $t_{n+1}$ DSM 29 $t_{n+1}$ DSM 47 $t_{n+1}$ DSM 9 $t_{n+1}$ DSM 9 $t_{n+2}$
	15 Sec.	Quadrant Time - solar Quadrant Time - backgrnd	DSM 22 DSM 24
	30 Sec.	Energy ID (Sync)-Low Band (7.5-15, 15-30 keV)  High Band (30-60, 60-120 keV)	DSM 19 $t_n$  DSM 19 $t_{n+1}$
Central Detector Neutron	200 Sec. 50 Sec. 25 Sec.	MeV - solar MeV - solar MeV - solar MeV - background MeV - background MeV - background	2 MF Word (11,12)  6 channels on 1 8 bit word
Front Slab	15 Sec.	>50 keV - solar >50 keV - background	ASSC 5 ASSC 9
Cup - C.P.	15 Sec.	>50 keV B >50 keV S	ASSC 6 ASSC 10



TABLE I (B)

Measurement	Time Resolution	Information	Readout Assignment
Housekeeping	m x 15.36 sec	1. Detector Temp	ASC 2 $t_n$
		<ul style="list-style-type: none"> <li>. Electronic Temp</li> <li>. Central HVPS - Analog</li> <li>. Shield HVPS - Analog</li> <li>. LVPS - Analog (1/5x20V)</li> <li>. { <ul style="list-style-type: none"> <li>command status</li> <li>magnetometer pulse enable</li> <li>auto radioactive source calibration</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>.</li> <li>.</li> <li>.</li> <li>.</li> <li>.</li> </ul>
		1. ID Status	ACS 1
		. Command status	.
		- Scan Mode	.
		. Command status	.
		- Quadrant and gain mode	.
		. Command status	.
		- Inflight calibration on/off	.
		. Command status	.
		- (2) HVPS on/off	.
		. Command status	.
		- HV Programmer	.

TABLE II

Command No.	BBRC Pin No. (Connector 09J02)	When Affected	Command
			Mode when power is turned on is under- lined.
1	U	EOS	Quadrant ( <u>Normal</u> / Alternate)
2	V	EOS	Scan ( <u>Full</u> , Inter- mediate, Short)
3	W	EOS	Calibrate (on or <u>off</u> )
4	X	EOS	Gain Change ( <u>X1</u> or X 1/3)
5	Y	EOS if issued once, n steps immediately for n+1 commands	Central HV Step ( <u>0</u> thru 63)
6	Z	n steps if EOS interrupts command sequence Immediate	Central HV (ON/ <u>OFF</u> )
7	<u>A</u>	Immediate	Shield HV (ON/ <u>OFF</u> )
8	<u>B</u>	Immediate	Auto Calibration (Inhibit/ <u>Active</u> )
9	<u>C</u>	Immediate	Magnetometer Pulse (Enable/ <u>Disable</u> )

EOS - End of scan: at completion of readout of both  
solar and quadrant for MF data.

All instrument circuitry predicated on single (non-  
redundant) pulses on command lines.

## II. EXPERIMENT OPERATION

The central detector measures the energy spectra of gamma rays passing through, without interacting in, the thin front slab. The thicker cup absorbs a fraction of the gamma rays produced within the spacecraft and also rejects a fraction of those gammas which merely scatter in the central detector. Both shields also reject charged particles. Gammas are recorded while the sun is within  $45^\circ$  of the compartment center line and similarly in the opposite direction for background comparison. An opening angle can only be defined in terms of the relative absorption of gammas in the thin front slab and thicker cup. A nominal value of  $45^\circ$  can be used.

The magnetic core memory is divided into two parts, so that one half can continuously record gamma rays of all energies while the other is being read out. When a readout is completed the roles of the two halves are reversed.

The operation of the experiment in the various modes can now be described by references to Figures 1 and 3 and Tables I and II. The various operational modes are:

- Normal Operation
- Night Operation
- Change Time Resolution
- Change Gain
- Change H.V.
- Quadrant Change
- Calibration

### Normal Operation:

In the normal mode of operation the data assigned to the main frame, DMF, DSF, and ASC words are previously stored in

solar and background buffers, logarithmic count meters, or adaptive accumulators (see Figure 3). Designation of an event in a given detector element for labeling as solar or background is determined first by the day azimuth aspect signal from the spacecraft. This signal is provided by the BBRC sun sensor as it sees the sun before the axis of our detector does ( $41.2^{\circ}$ ). This signal goes to a quadrant generator which generates a signal identifying the quadrant the detector lies in. If an event occurs in the allowed quadrant interval (1 or 3) it is appropriately recorded.

The various data is accumulated for a time interval dependent on the telemetry sampling rate, and this determines the time resolution listed in Tables I and III. For example, in the normal mode the central detector gamma ray spectrum is analyzed with a 377 channel quadratic analyzer, and the spectrum is accumulated over a real time of 184.32 sec (duty cycle 25% due to wheel spin, actual accumulation time is 46 sec. The quadrant time is monitored.)

In the normal mode of operation similar remarks apply to all other data listed in Table I.

#### Night Operation:

The experiment is intended to operate continuously through the night; however, the quadrant generator is now controlled by either the night azimuth signal (gyro) or the magnetic aspect (magnetometer) signal. In the former case, an apparent sun

pulse is generated so the two quadrants during which data is accumulated correspond to directions away from the sun (into the night sky) and toward the sun (thus intercepting the earth).

If, for some reason, it is desired to change from the normally chosen gyro-signal to the magnetometer signal for the night reference, a magnetic pulse enable command would be activated (see Table II). The detector viewing direction with respect to the earth sun line will, of course, be different than in the former case. The status of this command is monitored in the MF status word and ASSC 2-6. The day/night signal automatically causes the quadrant and generator to select the solar reference or night reference (which ever is enabled) signal.

#### Change Time Resolution:

The time resolution for obtaining the gamma ray spectrum may be reduced by using only a portion of PHA memory. Thus, by external command, time resolutions of 61.44 sec and 30.72 sec are available (normally 184 sec, see Table I and III). The available channels in this mode sample certain selected portions of the spectrum. The status of this command is monitored by analog word ASSC 1-2.

#### Change Gain:

The portion of the central detector spectrum analyzed may be selected by issuing a command which changes the gain of the central detector channel by 1/3. This is done by an appropriate change of the high voltage. The status of this command is monitored by analog word ASSC 1-3 and the DMF status word (bit 10).

TABLE III  
DATA TIMING FORMAT

TIME FRAME-SEC	MAIN FRAME			DSM	ASSC*
	FULL	INTER	SHORT		
1 15.36	(B) <sub>1</sub>	(B) <sub>1</sub>	(B) <sub>1</sub>	(S+B) <sub>1</sub>	<sup>1</sup> (S+B) <sub>1</sub>
2 30.72			(S) <sub>1</sub>		<sup>2</sup> (S+B) <sub>1</sub>
3 46.08		(S) <sub>1</sub>	(B) <sub>2</sub>	(S+B) <sub>2</sub>	<sup>3</sup> (S+B) <sub>1</sub>
4 61.44			(S) <sub>2</sub>		<sup>4</sup> (S+B) <sub>1</sub>
5 76.80		(B) <sub>2</sub>	(B) <sub>3</sub>	(S+B) <sub>3</sub>	<sup>5</sup> (S+B) <sub>1</sub>
6 92.16			(S) <sub>3</sub>		<sup>6</sup> (S+B) <sub>1</sub>
7 107.52	(S) <sub>1</sub>	(S) <sub>2</sub>	(B) <sub>4</sub>	(S+B) <sub>4</sub>	<sup>1</sup> (S+B) <sub>2</sub>
8 122.88			(S) <sub>4</sub>		<sup>2</sup> (S+B) <sub>2</sub>
9 137.24		(B) <sub>3</sub>	(B) <sub>5</sub>	(S+B) <sub>5</sub>	<sup>3</sup> (S+B) <sub>2</sub>
10 153.60			(S) <sub>5</sub>		<sup>4</sup> (S+B) <sub>2</sub>
11 168.96		(S) <sub>3</sub>	(B) <sub>6</sub>	(S+B) <sub>6</sub>	<sup>5</sup> (S+B) <sub>2</sub>
12 184.32			(S) <sub>6</sub>		<sup>6</sup> (S+B) <sub>2</sub>

NOTE: B = BACKGROUND (ANTI SOLAR DATA)

S = SOLAR

\*ASSC WORDS 1 & 2 SUBCOMMUTATED IN 6 FRAME CYCLES  
NOT NECESSARILY SYNCHRONIZED WITH DIGITAL OUTPUTS

High Voltage Change:

The operating high voltage on the central detector photo-multiplier may also be changed in increments by a step command. This voltage may be varied over approximately 200 volts in 64 steps. The step reference is reported by ASSC 1-6 and an analogue reading of the voltage is given by ASSC 2-3.

Quadrant Change:

The viewing quadrants for the detector are normally 1 and 3 (see Figure 2). A change to quadrants 2 and 4 may be made on command. The status of the command is monitored by analogue word ASSC 1-3 and the DMF status word (bit 2).

Calibration Mode:

Normally the instrument functions with the gamma ray and x-ray calibration lines electronically removed from the spectrum of the respective detectors. At the passage of the satellite from either night into day, or day into night, a radioactive source calibration is automatically initiated which lasts for three (3) minutes. Evidence for this operation is monitored on analogue word ASSC 1-4, and the DMF status word (bit 6). This automatic calibration sequence may be removed by activating the auto calibrate inhibit command.

A calibration sequence can also be initiated in real time by a calibrate on/off command to provide electronic and radioactive source calibration source data. Status of this command is monitored by analogue word ASSC 1-4, and the DMF status word (bits 5 and 6).

Emergency Operation:

In case of instrument or spacecraft difficulties, the following procedure should be followed:

1. If there is corona or arcing, or if the analog data on ASSC 2-3 or ASSC 2-4 show excessive high voltage or unusually low voltage, then the two high voltage off commands should be issued.

These alarm levels are:

Voltage ASSC 2-3

$1500 < E_o < 2200V$  for an EMR central detector

$1500 < E_o < 1900V$  for an RCA central detector

ASSC 2-4

$1000 < E_o < 1500V$  for C70132B

2. If the instrument is stuck in the auto calibrate mode, the Auto Calibrate Inhibit command should be issued. If the instrument is in normal command calibrate mode, issuing a second command returns it to normal operation.
3. Telemetry dropout: Turn off the high voltage automatically with the computer. If the high voltage is found not to be involved in such dropouts, this feature should be disabled.



### III. DATA FORMAT

In reporting this data format we will make the following assumptions:

1. The primary use of this description will be during checkout at Ball Brothers.
2. Ball Brothers has a computer which can receive key board or sense switch commands, and display various segments of the data we wish displayed.
3. The data format should be that which makes the most sense to us as experimenters.

For our instrument, the DMF, DSF, and ASSC data are independent of each other, and so we will discuss them separately in the following.

#### A. Main Frame Data:

Our primary data, that is the data from our NaI(Tl) detector, comes out on main frame words 11 and 12. Data is taken in two quadrants, one centered on the solar direction and the other on the anti-solar direction. These are referred to as solar and background quadrants, respectively. There is a mode in which the two other quadrants are chosen instead of these. This is referred to as the alternate quadrant mode. The main portion of our data is essentially a readout of a pulse height analyzer. There are three time scales involved. There is the short scan in which the background and solar data is readout in approximately 31 seconds; the intermediate scan, taking 61 seconds; and the full scan, taking 184 seconds. The speed of the I/O devices available at Ball Brothers will determine to some degree how

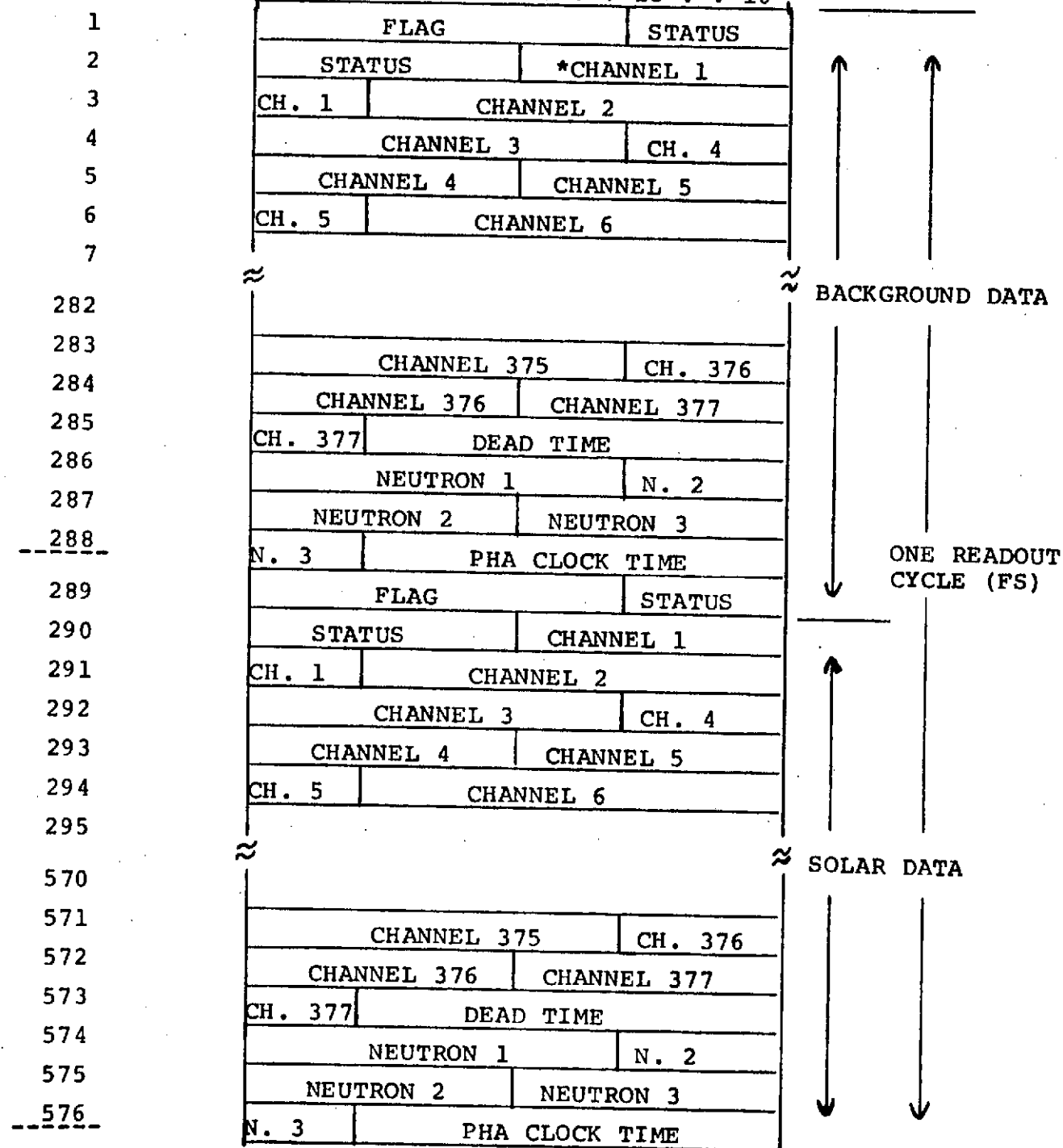
our data is displayed. For example, if a printer is available that can print approximately four (4) lines per second, our best display would probably be a histogram type graph.

The formats for the full, intermediate, and short scans are shown in Figures 4, 5, and 6. The internal instrument word length is 12 bits. The main frame word length is 8 bits, the main frames 11 and 12 are linked together for a total of 16 bits. The consequence is that instrument words are chopped up between main frame words 11 and 12; however, four instrument words fit into six main frame words or into three main frame cycles. Each scan, whether from background or solar quadrant begins with a 12 bit flag of one's. This is followed by a status word shown in Table IV. Bits 3 and 4 of the status word indicate the scan mode of the next data block to be read out. The words following the status word give the number of counts in each subsequent channel of the pulse height analyzer. In the intermediate and short scan modes not every channel of the analyzer is covered, just certain selected channels (Table V). The last five instrument words in each quadrant readout give data on neutrons, dead time, and accumulation time for the pulse height analyzer. The two low order bits in the status word are the least significant bits for the dead time given in the fifth from the last word. For each quadrant readout, the status should be displayed in a mnemonic form, rather than just displaying bits. The number of counts in each channel should be displayed in decimal. If the printer is fast enough, a graphical display probably would be more easily interpreted. The number of counts in the three neutron

# MAIN FRAME FORMAT - FULL SCAN\*

19.

Main Frame



\* FOR ENERGY-CHANNEL ASSIGNMENTS  
SEE GAMMA CHANNEL TABLE

FIGURE 4

## MAIN FRAME FORMAT - INTERMEDIATE SCAN\*

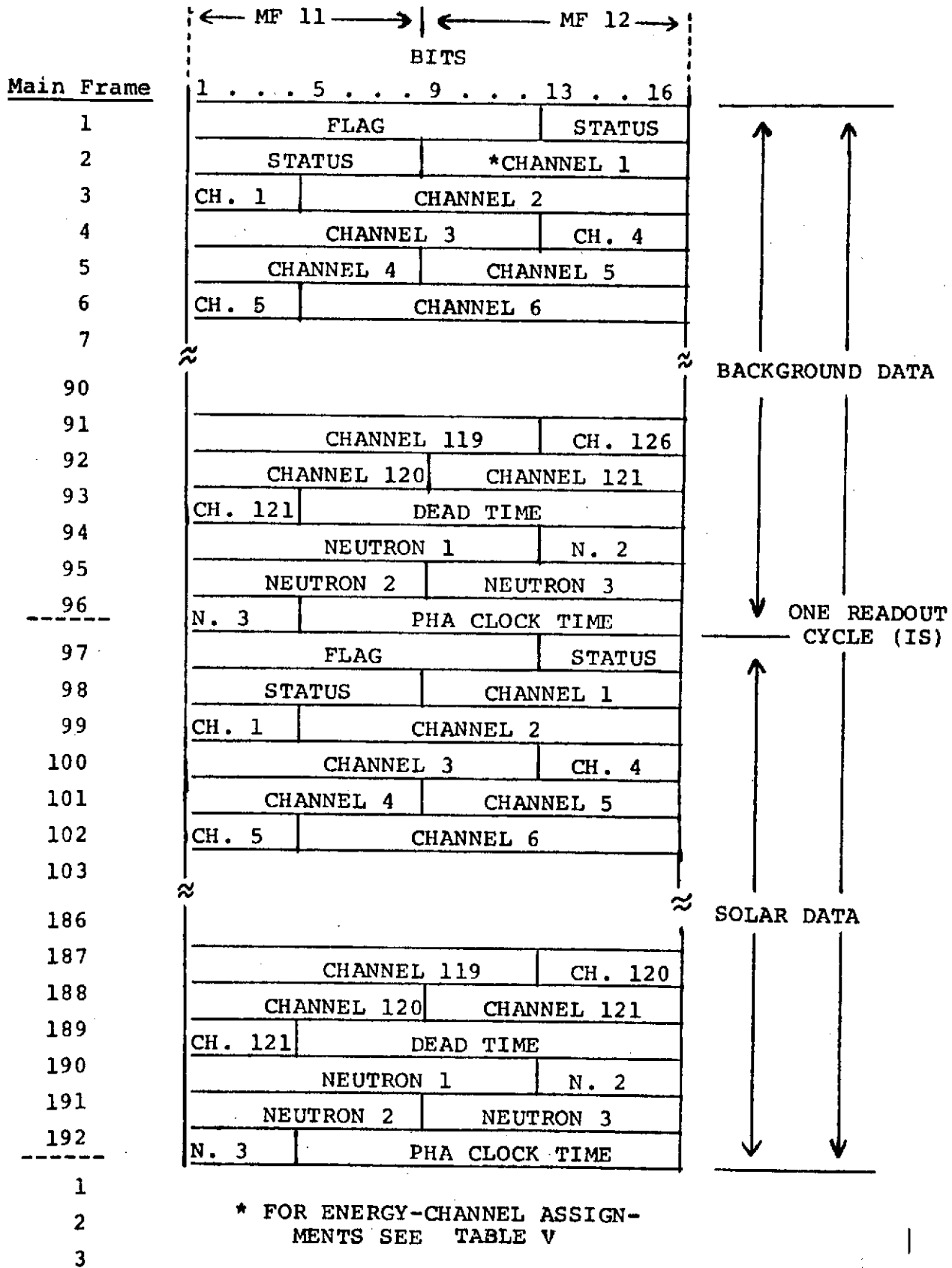


FIGURE 5

MAIN FRAME FORMAT - SHORT SCAN\*

Main Frames

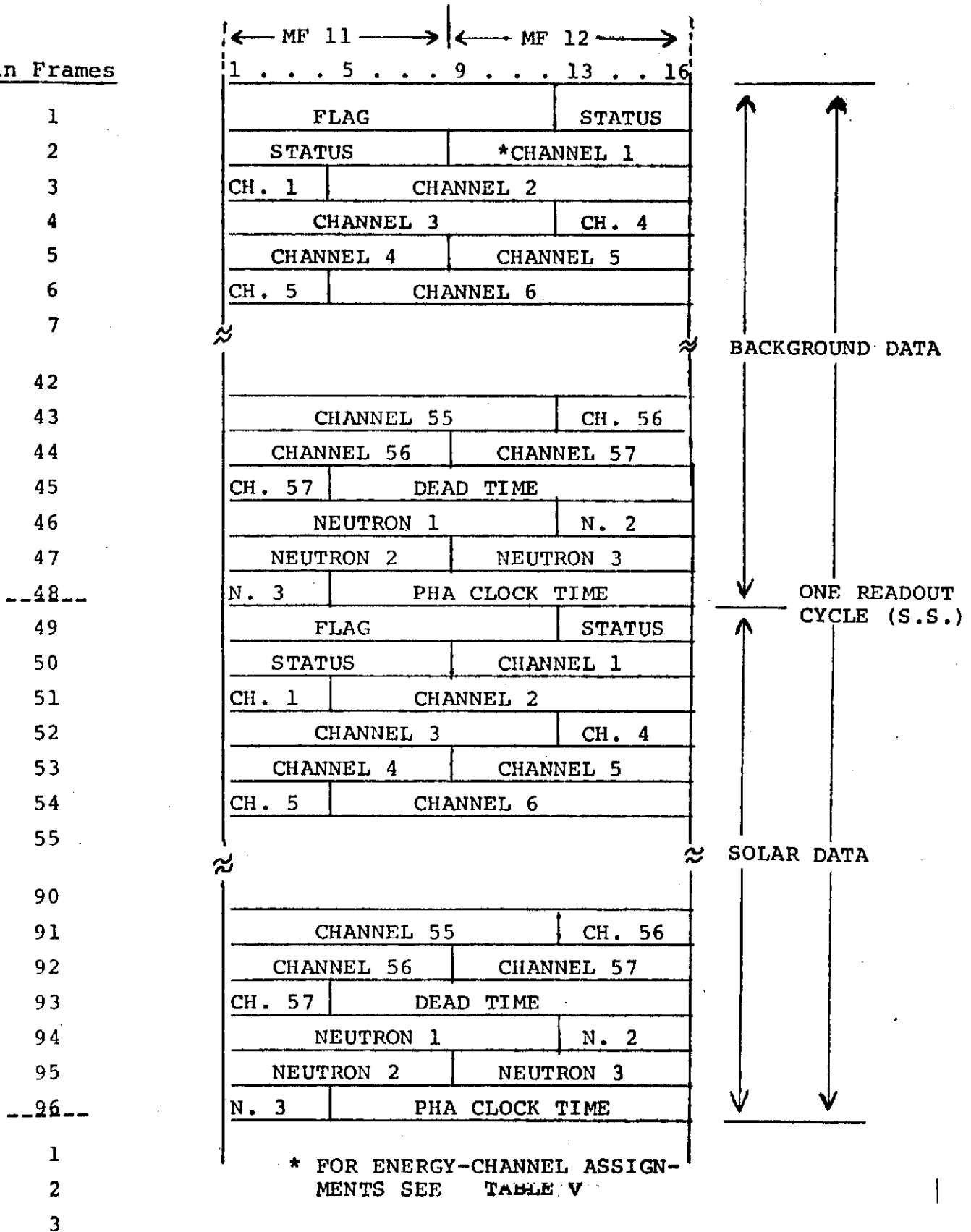


FIGURE 6

TABLE IV.

Main Frame - Experiment Words: 12 Bits. Total Experiment

Words Per Cycle:  $2 \times 384 = 768$ 

MF Flag: 12 "1's"

MF Status:	S1	1	Solar Data
		0	Background Data
	S2	1	Normal Quadrant Mode
		0	Alternate Quadrant Mode
	S3 & S4	S4    S3	
		0    0	Full Scan Mode
		1    0	Intermediate Scan Mode
		0    1	Short Scan Mode
	S5	1	Electronic Calibration ON
		0	Electronic Calibration OFF
	S6	1	Radiation Source Calibration ON
		0	Radiation Source Calibration OFF
	S7	1	Day
		0	Night
	S8	1	Magnetometer Reference
		0	Gyro Reference
	S9	1	Charge Particle HVPS ON
		0	Charge Particle HVPS OFF
	S10	1	Gain = Times one
		0	Gain = Times one third
	S11	(dead time)	Least significant bit
	S12		Next least significant bit

Channel Data:

Most significant bit FIRST

Least significant bit LAST

Channel 1 is lowest energy window

Channel 377 is highest energy window

PHA Live Time:

1 bit =  $[1/390.625]$  seconds x 8 Full S.

2 Inter S.

1 Short S.

PHA Dead Time:

1 bit = 1.495 msec.

TABLE V.  
Gamma Channels (in time of full scan channel numbers)

Full Scan word no.	Intermediate scan word no.	Short scan word no.
1	1	1
9	2	---
17-33	3-19	2-18
41	10	---
49	21	19
57	22	---
65	23	20
73	24	---
81-97	25-41	21-37
105	42	---
113	43	38
121	44	---
129	45	39
137-153	46-62	40-56
161	63	57
169	64	
177	65	
185	66	
193	67	
201	68	
209	69	
217	70	
225	71	
233-249	72-88	
257	89	
265	90	
273	91	
281	92	
289-305	93-109	
313	110	
321	111	
329	112	
337	113	
345	114	
353	115	
361	116	
369-373	117-121	

channels should be displayed separately, and the pulse height analyzer dead time and accumulation time should be scaled according to the appropriate factor, to be determined, and given in seconds. The University of New Hampshire has developed Fortran and IBM Assembler Language programs which may be of some value in reformatting this data.

#### B. DSF Data Format:

The basic format is shown Table VI. The DSF data is primarily concerned with the x-ray detector. The counts in each detection range, for example 7.5 to 15 keV background data, are not true counts, but counts compressed by an adaptive scaler. These counts may be decompressed according to the following function:

$$\begin{aligned}
 C &= S & S < 95 \\
 &= 4 (S-96) + 98 & 96 < S < 127 \\
 &= 16 (S-128) + 232 & 128 < S < 159 \\
 &= 64 (S-160) + 752 & 160 < S < 191 \\
 &= 256 (S-192) + 2784 & 192 < S
 \end{aligned}$$

Where C is the corrected count number and S is the value of the scaler.

The appropriate scale factor for the quadrant time (accumulation time for the x-ray scalers) will also be provided.

#### C. Analog Subcom Data Format:

The basic formats are shown in Tables VII, VIII, and IX. The University of New Hampshire is assigned analog subcom words 1-10. We have sub-subcommed analog words 1 and 2. Analog subcom



TABLE VI  
DSF DATA FORMAT

DSM WORD NO.		(MF 17)	
1	19		Energy ID (Low) all zeros (0)
2	22		Quadrant Time - Solar
3	23	7.5-15	keV data - Solar
4	24		Quadrant Time - Background
5	29	7.5-15	keV Data - Background
6	47	15-30	keV Data - Solar
7	9	15-30	keV Data - Background
8	19		Energy ID (High) all ones (1)
9	22		Quadrant Time - Solar
10	23	30-60	keV Data - Solar
11	24		Quadrant Time - Background
12	29	30-60	keV Data - Background
13	47	60-120	keV Data - Solar
14	9	60-120	keV Data - Background

1

2

3

Low Energy ID (7.5-30) All 0's

High Energy ID (30-60 keV) All 1's

Complete x-ray cycle takes two DSF cycles or 31.32 seconds.

TABLE VII.  
ASSC DATA FORMAT

ASSC Word (MF 1)

 BBRC Pin No.  
 (connector #09J01)

1 - 1	Sync	COMMAND STATUS	M
2	Scan Mode		
3	Quadrant and Gain Mode		
4	IFC ON/OFF		
5	HVPS's ON/OFF		
6	HVPS Programmer Step		
2 - 1	Detector Temperature	HOUSEKEEPING AND COMMAND STATUS	N
2	Electronic Temperature		
3	Central HVPS Analog		
4	Charged Particle HVPS Analog		
5	LVPS Analog		
6	Mag Pulse Enable		
3	$\gamma$ .3 to 9.1 MeV	Background .....	Q
4	$\gamma$ > 9.1 MeV	Background .....	R
5	CP Slab	Background .....	S
6	CP Cup	Background .....	T
7	$\gamma$ .3 to 9.1 MeV	Solar .....	U
8	$\gamma$ > 9.1 MeV	Solar .....	V
9	CP Slab	Solar .....	W
10	CP Cup	Solar .....	X

TABLE VIII.  
ASSC DATA SUMMARY - ASSC 1-n  
COMMAND STATUS

n

1. FLAG (ID Sync) 5.0 volts
2. SCAN MODE
  - (00) Full Scan = .05 volts
  - (10) Intermediate Scan = 0.9 volts
  - (01) Short Scan = 1.8 volts

## 3. QUADRANT AND GAIN MODE

Quadrant	Gain	Output
(0) Alternate	(0) Times 1/3	0.05 V
(1) Normal	(0) Times 1/3	0.9 V
(0) Alternate	(1) Times 1	1.8 V
(1) Normal	(1) Times 1	2.7 V

4. IFC MODE
  - (00) IFC OFF = 0.05 Volts
  - (10) Electronic Calib. = 0.9 Volts
  - (01) Radiation Source Calib. = 1.8 Volts

## 5. HVPS MODE

Central HVPS		Charge Part. HVPS		Output
0	OFF	0	OFF	0.05
1	ON	0	OFF	0.9
0	OFF	1	ON	1.8
1	ON	1	ON	2.7

## 6. HVPS REFERENCE 64 steps

Gain x 1/3	0.000 - 2.112	} 33 mv/step
Gain x 1	1.500 - 3.162	

TABLE IX.  
ASSC DATA SUMMARY - ASSC 2-n  
HOUSEKEEPING

- n
1. Temperature - Detector  $+50^{\circ}\text{C}$  = 1.55 V (nominal)  
 $0^{\circ}\text{C}$  = 3.25 V (nominal)
  2. Temperature - Electronics  $-30^{\circ}\text{C}$  = 4.27 V (nominal)
  3. Central HVPS Analog  $\approx$  4 V estimate
  4. Charged Particle HVPS Analog  $\approx$  4 V estimate
  5. LVPS Analog 1 -  $20 \text{ V} \div 5 \approx 4 \text{ V}$
  6. Magnetometer Aspect Pulse Enable and Auto Radio-  
active Source Calibration Inhibit

AUTO RS CAL.	MAG. ASPECT	OUTPUT
OFF	OFF	0.05V
ON	OFF	0.9
OFF	ON	1.8
ON	ON	2.7

word 1 displays a flag word valid for both ASSC words 1 and 2, and various command status words. ASSC 1-2, 1-3, 1-4 contain redundant information also given in the MF status word. For analog subcom word 2, the first two sub-subcom words give temperature information, words 3 and 4 give analog information for the two high voltage power supplies, word 5 monitors the low voltage power supply, and word 6 indicates the command for auto RS calibration and magnetometer aspect pulse enable. Analog subcom words 3-10 monitor counting rates in various parts of the instrument. Calibration graphs will have to be provided for voltage versus counting rate for these various rates. This will be done experimentally during calibration of the instrument.

#### Calibration Formats:

For both the command calibration and automatic calibration operations the commands are held up until the completion of a readout. The calibration sequences were designed for the full scan mode and therefore should be normally initiated only in that mode.

The command calibration initiates an electronic calibration sequence, lasting 184 seconds, during which time discriminator thresholds are tested with pulses of various voltages. The number of pulses differs for each test voltage so they may be distinguished in the spectral readout. The input pulse rates and their relation to the discriminator settings are summarized in Table X.

Table XI illustrates the outcomes for testing the central detector electronics.

TABLE X.

## ELECTRONIC IFC TIMING (FULL SCAN)

EOS

← 15 sec → |

1		2		3		4		5		6		7	
N		N		*		*		*		N		N	
9.1+MeV		0.3-MeV		0.3+MeV		1.0 MeV		9.1-MeV		9.1+MeV		0.3-MeV	
781 cps		48 cps		6.25 kc		781 cps		48 cps		6.25 kc		781 cps	
T <sup>-</sup>		T <sup>-</sup>		T <sup>-</sup>		T <sup>-</sup>		T <sup>-</sup>		T <sup>+</sup>		T <sup>+</sup>	
6.25 kc		781 cps		100 kc		6.25 kc		781 cps		100 kc		6.25 kc	
N		N		N		N		N		*		*	
7.5-15	15-30	30-60	60-120	7.5-15	15-30	30-60	60-120	7.5-15	15-30	30-60	60-120	7.5-15	15-30
U <sup>+</sup>	L <sup>+</sup>			U <sup>-</sup>	L <sup>-</sup>			U <sup>-</sup>	L <sup>-</sup>			U <sup>+</sup>	L <sup>+</sup>
N	*	N	N	*	N	N	N	*	N	N	N	N	N
												(Coinc Re- ject)	
← *		← *		← *		← *		48 Hz *		N		N	

Definitions:

AG = Energy Window

T = Threshold of Cup or Slab

\* = System Data Acceptance

N = System Data Not Accepted

+ = + 10% (nom.)

- = - 10% (nom.)

V<sub>T0</sub> = Test Voltage for Cup/SlabU = Upper Discriminator Threshold for X-ray Detector  
Attenuated for Given RangeL = Lower Discriminator Threshold for X-ray Detector  
Attenuated for Given Range

8		9		10		11		12		CYCLE NUMBER
N (coinc. reject)		N (coinc. reject)		*		N		N		MAIN CHANNEL
0.3+MeV		1.0 MeV		9.1-MeV		9.1+MeV		0.3-MeV		
48 cps		6.25 kc		781 cps		48 cps		6.25 kc		
T <sup>+</sup>		T <sup>+</sup>		T <sup>-</sup>		T <sup>-</sup>		T <sup>-</sup>		
781 cps		100 kc		6.25 kc		781 cps		100 kc		CUP/SLAB
*		*		N		N		N		
30-60	60-120	7.5-15	15-30	30-60	60-120	7.5-15	15-30	30-60	60-120	X-ray at 6.25 kc for all cases
N	N	U <sup>-</sup>	L <sup>-</sup>	N	N	U <sup>-</sup>	L <sup>-</sup>	N	N	relation to discriminator setting
		(Coinc Re-ject)								
N		N		*		*		*		neutron channel two (20 to 30 MeV Channel) 48 cps all cycles

EOS = Leading Edge of DMF Word 32, following WASC Word 48 and Last DMF Words 11, 12 which output Solar Data

TABLE XI.

## MAIN FRAME ELECTRONIC CHECK

Nominal Channel	Nominal Energy	Approximate no. of pulses generated $X-1/4$	Exact * Equation	Interpretation if observed
-3	.270 kev	2577(OV6) $\pm 3000$	7079*T	Lower threshold set too low
6	.330 kev	3520(OV5) $\pm 3000$	6250*T	Normal (lower threshold <del>not too high</del> )
			629 $P_4$ T	Coinc. reject may not be operating
70	1.0 MeV	2999 $\pm 360$	781*T	Normal (1 MeV location)
		4119(OV5) $\pm 3000$	7031*T	Coincidence's reject not working
354	8.19 MeV	3183 $\pm 415$	829*T	Normal (upper threshold not too low)
400	10.0 MeV	2577(OV6) $\pm 3000$	7079*T	Upper threshold set too high

OVn - Counts have overflowed scaler n times.

T =  $\frac{\text{clock time}}{4}$

\* to be calculated modulo 4096

T =  $\frac{15.36}{4}$  for each cycle.



The cup and slab thresholds and log rate meters for cup, slab, and central detectors are tested in 15.36 second segments; that is, the test frequencies are changed after every 48 ASSC words. The expected rates are given in Table XII. "Cycle number" indicates which group of 48 ASSC words the expected rate is to be found in. Counts in ASSC 3-10 in any cycles other than indicated are anomalous and should be printed out for diagnosis.

The twelve cycle group will start immediately after a solar quadrant has been read out if a command has been issued. To be specific, the calibration command is executed at the trailing edge of the DMF sync (i.e. the end of DMF word 32) following the readout of solar clock time. The twelve ensuing 15.36 second intervals constitute the cycle group referred to. This should allow the calibration sequence to be identified in real time. The sequence may always be identified retrospectively from the data.

The circuitry for the X-ray detector consists of an upper and lower discriminator preceded by a variable attenuator which is stepped to give various energy ranges. The disc thresholds are checked with pulses  $\pm 10\%$  above and below their nominal values. Table XIII shows the only places which should contain data.

The radioactive calibration format is handled as routine data.

TABLE XII  
COUNT RATE CHECK

ASC #	FUNCTION TESTED	EXPECTED VOLTAGE RATE (48 Hz)	ASC CYCLE #		
			(781 Hz) V	(6.25k Hz) V	(100k Hz) V
3	gamma rate (.3-9)B	5	4, 10	3	----
7	gamma rate (.3-9)S	5	4, 10	3	----
4	gamma rate (>9)B	11	1	6	----
8	gamma rate (>9)S	11	1	6	----
5	slab rate B	----	8	7	6, 10
9	slab rate S	----	8	7	6, 10
6	cup rate B	----	8	7	6, 10
10	cup rate S	----	8	7	6, 10

B - Background

S - Solar

cycle # explained in text

TABLE XIII.

## X-RAY CHANNELS WHICH SHOULD CONTAIN DATA

<u>Cycle #</u>	<u>DSF #</u>	<u>Description</u>	
1	9	15-30 keV	B
	47	15-30	S
3, 5, 11	23	7.5-15	S
	29	7.5-15	B

#### IV. SPECIAL HANDLING REQUIREMENTS

The instrument will be packed in a special case for transport to BBRC and ETR. It should be handled as a delicate scientific instrument. During storage the case will be flushed and pressured slightly above ambient with dry nitrogen. The temperature and humidity restrictions are:

< 40% relative humidity

< 10°C per hour temperature rate of change

#### V. ENVIRONMENTAL REQUIREMENTS

Even though the instrument is sealed, it has not been designed for high humidity. The scintillation crystals are hygroscopic so it is desirable to limit high humidity and direct water exposure to the instrument.

The detector should be kept in an environment with relative humidity of less than 40% at all times.

The detector should not see a temperature change of more than 10°C/hour. Therefore, prudence should be used in tests at the magnetics building and the open solarium. This figure for the temperature changes is that recommended by Harshaw Chemical Company for the NaI(Tl) central detector. The thermal inertial of the complete assembled detector may modify this figure for the instrument as a whole. This can be determined during testing since the temperature of the central detector is monitored by ASSC 2-1.

At ETR the humidity in the compartment could be controlled by circulating dry N<sub>2</sub> around the instrument.

## VI. SPECIAL TEST REQUIREMENTS

Gamma ray angular distribution measurements will be required when all experiments are in spacecraft. See also VII and VIII.

## VII. ALIGNMENT REQUIREMENTS

Alignment of front slab cover to be perpendicular to compartment center line to  $1/2^\circ$  is adequate.

## VIII. RADIOACTIVE SOURCE REQUIREMENTS

We need small (1 to 10  $\mu\text{c}$ ) sources to verify operation of our detector during tests. Also we would like to measure the spacecraft attenuation using a strong source and rotating the spacecraft in both pitch and azimuth. This should be done with a full complement of instruments.

A possible method of checking the quadrant generator is shown in Figure 7. A quadrant signal will also be available from the instrument. This will more probably be used.

In addition the instrument contains two permanently mounted sources,  $\text{Am}^{241}$  and  $\text{Co}^{60}$ . The strength of each is approximately 0.02  $\mu\text{c}$ .

A long run with the instrument supported externally, but located in the spacecraft compartment should be made. The spacecraft should be wheeled away and the background subtracted. This will give some information about the activity of the spacecraft itself.

#### IX. HIGH VOLTAGE OPERATION REQUIREMENTS

Instrument will operate at atmospheric pressure. In vacuum testing the high voltages should be turned off during pump down.

#### X. SQUIB REQUIREMENTS

None.

# XI. TEST PROCEDURES AT BBRC

## 1. Exercise of Commands

Each command should be exercised singly. Combinations and permutations test the internal system logic and are not likely to cause problems in the s/c command transmission channels.

<u>Command</u>	<u>Verification</u>
Quadrant	Repositioning of source if s/c is spinning and print out of spectrum, or observing electronic quadrant signal from instrument.
Gain change	Printout of spectrum and status (ASSC 1-3)
Magnetometer pulse Enable	Repositioning of source and dummy magnetic pulses (e.g. 90° to gyro signal)
Mode Command	Printout of spectrum
Calibration on/off	Printout of spectrum, DSM, and ASC
- HVPS step command	Issue command n times. Print spectrum housekeeping before and after
- HVPS on/off command	Print spectrum and housekeeping
Shield HVPS on/off	Print spectrum and housekeeping before and after
Auto R.S. Calibration Inhibit	Print Spectrum

2. Some sort of source test; turn HVPS off and run for long time. (Other EE type tests?) Spin rate at minimum and maximum.

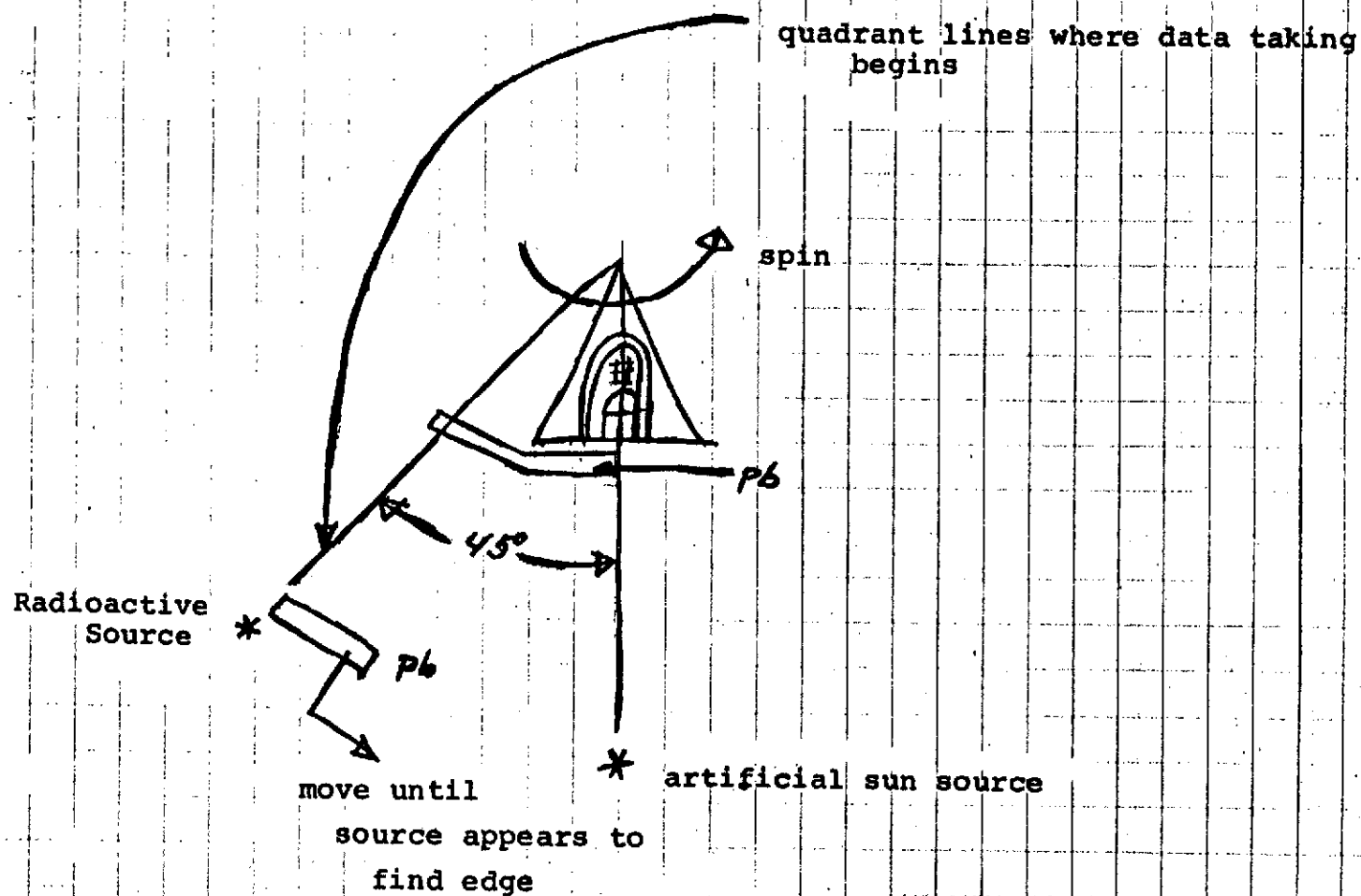


FIGURE 7  
quadrant generation check



# FIGURES

<u>Title</u>	<u>Page</u>
1. Detector Schematic.....	3
2. Arrangement of Detector in OSO. Compartment Showing Quadrants.....	5
3. System Functional Block Diagram.....	6
4. Main Frame Format - Full Scan.....	19
5. Main Frame Format - Intermediate Scan.....	20
6. Main Frame Format - Short Scan.....	21
7. Quadrant Generation Check.....	39

46  
41.

## TABLES

<u>Title</u>	<u>Page</u>
I. Data Assignment	
A. MF and DSF Data .....	8
B. Housekeeping.....	9
II. Command Assignment.....	10
III. Data Timing Format.....	14
IV. MF Status Word.....	22
V. Gamma Channels for Short and Intermediate Scans.....	23
VI. DSF Data Format.....	25
VII. ASSC Data Format.....	26
VIII. ASSC 1 Sub-subcom Format.....	27
IX. ASSC 2 Sub-subcom Format.....	28
X. Summary of Electronic Test Pulses.....	30
XI. Main Frame Electronic Check.....	31
XII. Count Rate Check.....	33
XIII. X-Ray Threshold Check.....	34